2002年6月1日発行（毎月1回1日発行）第21巻第6号 昭和57年12月6日第三種郵便物認可

マイコン

わかる！動かせる！
プログラムが組める雑誌

BASIC Magazine

JUN. 2002 6

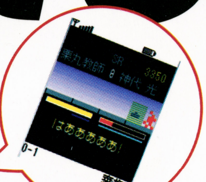


定価 550Yen

掲載プログラムがダウンロードできる！ http://www.basicmagazine.net/

特集 携帯電話で動くプログラムに挑戦！！

君にも作れる「iアプリ」





21世紀をリードする
技術者育成のための独特なカリキュラム
東京都立科学技術高等学校

BM基礎講座 ベーシックプログラミング教室
サブルーチンとキー入力

見やすい、楽しい、誰でも使えるサイトを実現！
一歩先ゆくホームページ作り
リンク—ページとページをつなげる、よそのホームページとつなげる

はじめようMicrosoft OfficeでBASICプログラミング
画面をつくろう

Visual Basicアルゴリズム講座
加速度の計算をしてみよう

チャレンジ！ PIC-BASIC
ワンチップマイコンでBASICインタプリタが動く

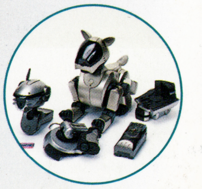
特選投稿プログラム・コーナー

Visual Basic V6.0	Action
Visual Basic V6.0	Move
Visual Basic V6.0	METEO
Visual Basic V6.0+DirectX	BIKE
VBScript + DirectX	くるくるアイ
Excel(VBA)	CLMS
F-BASIC V6.0	ミニゲーム9
HSP V2.55	go zigzag
iアプリ	i.Doll
P I ECE	連珠





パソコンで作ろう！ぼくらの夢のまち
インテル春休みPCスクール




パソコンも
一年たったら
ただの箱。
AMD
Duron
PROCESSOR
1GHz搭載
だったら初めは、
¥39,800

Easy 300　企業のネットワーク環境に最適なクライアントPC



※ディスプレイは別売りです。

AMD Duronプロセッサ 1GHz 搭載

**ワープロや表計算などビジネス用途から
ゲームなどのホビー用途まで使えるデスクトップPC**

■AMD Duronプロセッサ 1GHz
■SD-RAM 128MB PC133対応メモリ ■20.0GB Ultra-ATA100対応 HDD
■MSI製 VIA Apollo KLE133チップセット搭載マザーボード
■VIA Apollo KLE133内蔵ビデオアクセラレーター■最大40倍速CD-ROMドライブ
■AC97 3Dサウンド(オンボード) ■3.5インチ2mode FDD■MicroATXタワーケース
■10/100Base-TX LAN ■PS/2日本語キーボード ■PS/2スクロールマウス
■USBポート×2■1年間無償保証
※56Kbpsモデム追加 プラス¥4,500 ※15インチ液晶ディスプレイ プラス¥39,500

LANで差が出る。驚異のコストパフォーマンス
Easy 300 5台セットモデル
10M 5ポートHUB付き　セット価格　¥199,800

AMD Duron 1GHz モデル　**¥39,800**
WindowsXP HomeEdition インストールモデル　**¥53,300**

DeskNote C1100XL　オフィス向け低価格デスクノート



Intel Celeron プロセッサ 1.1GHz搭載

**デスクトップの性能をノートブックサイズに凝縮。
高性能かつ省スペース・機動性を発揮します。**

■Intel Celeronプロセッサ 1.1GHz ■SD-RAM 256MB PC133対応メモリ
■10.0GB Ultra-ATA100対応 HDD ■SiS630ST内蔵グラフィック(シェアメモリ)
■最大24倍速CD-ROMドライブ ■AC97 3Dサウンド(オンボード)
■14.1型TFT液晶ディスプレイ(最大解像度1024×768)
■10/100Base-TX LAN ■内蔵56Kbpsモデム
■サイズ(W)320×(D)260×(H)39mm ■重量 2.5kg
■USB×4/シリアル×1/パラレル×1/ディスプレイ出力×1/Sビデオ出力×1
■AC電源対応(バッテリーなし) ■1年間無償保証

Windows XP インストールモデル登場
DeskNote C1100XL WindowsXPモデル
Microsoft WindowsXP HomeEditionインストール
24倍速CD-ROMドライブモデル ¥113,300

Intel Celeron 1.1GHz 24倍速CD-ROMドライブモデル　**¥ 99,800**
Windows XP HomeEditionインストールモデル　**¥113,300**

m-Book C1200XL　ビジネスに必要な機能を搭載したA4ノートPC



Intel Celeron プロセッサ 1.2GHz搭載

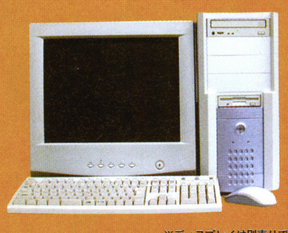
**CD-RW+DVD-ROMコンボドライブ搭載A4ノートPC
デスクトップのパフォーマンスを実現します。**

■Intel Celeron プロセッサ 1.2GHz ■SD-RAM 128MB PC133対応メモリ(最大512M
■20.0GB Ultra-ATA100対応 HDD ■SiS630S内蔵グラフィックシェアメモリ
■CD-RW+DVD-ROMコンボドライブ (CD-Rx4倍速/CD-RWx4倍速/DVDx6倍速)
■14.1型TFT液晶ディスプレイ(最大解像度1024×768ピクセル) ■3.5インチ2mode FDD
■10/100Base-TX LAN ■内蔵56Kbpsモデム (LANと排他処理)
■サイズ (W)308×(D)254×(H)37.5mm ■重量 3.0kg ■TypeII x1 CardBus対応
■IEEE1394×1/USB×2/PS/2ポート×1/パラレル×1/ディスプレイ出力×1
■リチウムイオンバッテリー(標準2.0時間/最大3.0時間) ■1年間無償保証

Windows XP インストールモデル登場
m-Book C1200XL WindowsXPモデル
Microsoft WindowsXP HomeEditionインストール
24倍速CD-ROMドライブモデル ¥123,300

Intel Celeron 1.2GHz 24倍速CD-ROMドライブモデル　**¥109,800**
CD-RW+DVD-ROMコンボドライブモデル　**¥129,800**

MDV-P2400GT/GF4MX　処理性能重視のハイエンドPC



※ディスプレイは別売りです。

Intel Pentium4 プロセッサ2.4GHz 搭載

**最新ビデオカードGeForce4 MX440L/DDR64MB搭載。
高速・大容量80GB 7200回転 ATA133対応ハードディスク採用モデル**

■Intel Pentium4プロセッサ 1.6AGHz/2.4GHz
■DDR SD-RAM512MB PC2100対応メモリ(256MB×2枚)
■80GB Ultra-ATA133対応 7200回転 HDD ■Intel 845D搭載マザーボード
■nVIDIA GeForce4 MX440L/DDR64MB AGPビデオカード
■Burn-Proof機能搭載40倍速CD-RWドライブ (40倍速CD-R/12倍速CD-RW/48倍速CD-ROM/ライティングソフト付属)
■16倍速DVD-ROMドライブ (16倍速DVD-ROM/40倍速CD-ROM/DVD再生ソフト付属)
■Creative SoundBlaster Live! Value(5.1chスピーカー対応)
■Promise IDE-RAIDコントローラ(RAID 0/1対応)オンボード ■3.5インチ2MODE FDD
■ATXミドルタワーケース (350W電源) ■112日本語キーボード
■PS/2スクロールマウス ■USB:背面×2 (USB2.0対応)/前面×2 ポート ■1年間無償保証

Windows XP インストールモデル登場
MDV-P2400GT/GF4MX WindowsXPモデル
Microsoft WindowsXP HomeEditionインストール
Pentium4 1.6AGHzモデル ¥143,300

Intel Pentium4 1.6AGHz モデル　**¥129,800**
Intel Pentium4 2.4GHz モデル　**¥189,800**

**詳しい製品情報やご注文は
インターネットかお電話でどうぞ**

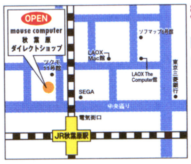
http://www.mouse-jp.co.jp/

Tel
048-760-1600/1601
電話受付時間：10:00〜19:00

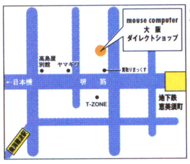
Fax
048-763-1177
24時間受付

20台以上ご購入のお客様［法人専用ダイヤル］
03-5209-3491
電話受付時間：10:00〜19:00

その性能に触れてお確かめください。



秋葉原ダイレクトショップ
■東京都千代田区外神田1-2-4
■JR秋葉原駅より徒歩2分
ツクモ11号館横
■営業時間：10:00〜19:00
水曜定休
TEL.03-5209-3474



大阪ダイレクトショップ
■大阪府大阪市浪速区日本橋4丁目8-8
小野ビル1F
■南海難波駅より徒歩5分
■営業時間：11:00〜20:00
年中無休
TEL.06-4396-6311

複数台購入割引実施中！
20台以上ご購入をお考えの方には、特別割引価格にてご提供中です。また御社ブランドPCのOEM供給も承ります。お問い合わせはお電話にてどうぞ。

通信販売のご案内
価格・在庫は常に変動しています。
ご注文の際にご確認ください。
お支払い方法は下記のいずれかをご利用ください。
※ご注文後のキャンセルは手数料2割を頂戴します。
●銀行振込（電信扱いにて）●現金書留
●通販分割（業界最低金利）●コンビニ支払い
●代金引換払い（コレクト）●各種カード決済
●60回リース
MDV全シリーズ1年間保証・初期10日間ピックアップ
送料はご負担いただきます。（着払い受け取りは出来ません。）
**振込口座　三井住友銀行　春日部支店
（普）7206485　（株）エムシージェイ**

●製品の表示価格には消費税は含まれておりません。また送料は別途申し受けます。●製品写真は出荷時のものと異なる場合があります。●Intel、Intel Insideロゴ、Intel Pentium、Intel CeleronはIntel Corporationの登録商標です。●Windowsは米国マイクロソフト社の米国及びその他の国における登録商標です。●その他の社名および製品名は各社の商標または登録商標です。

mouse computer
Get the Ideal
マウスコンピュータージャパン
〒344-0062 埼玉県春日部市粕壁東1-21-21
株式会社エムシージェイ

マイコンBASIC Magazine
http://www.basicmagazine.net

6 2002 JUNE





BASIC Magazine JUNE 2002 CONTENTS



© マイコンBASICマガジン6月号
第21巻 第6号 2002年5月25日印刷
2002年6月1日発行
定価 550円(本体524円)
電波新聞社 2002 Printed in Japan
編集発行人:平山哲雄
東京都品川区東五反田1-11-15
郵便番号 141-8715
電話　東京(03)3445-6111(大代表)
振替　(東京)00150-3-51961
印刷:奥村印刷(株)　製本:(株)堅省堂

・7月号は6月8日発売です!・

広告索引



MCG部，ロボット部など図書館に集合

編集部の学校訪問記

2001年4月に開校，今年2年目を迎えたばかりの科学技術高等学校（三上 勝校長）は，科学技術に関するスペシャリストを養成するための新しいタイプの高校です。都立高校で初めて「科学技術科」が設置されているなど，これまでの工業系高校とは違った独創的な専門科目が置かれています。同時に普通科科目も充実しており，大学などへの進学も大きな柱にしていることから，数学・理科・英語の単位数などに重点が置かれています。

今月号では，新設高校で学ぶ生き生きとした生徒たちの活躍を，MCG部やロボット部のクラブ活動，補習授業などを通して，紹介したいと思います。

東京都立科学技術高等学校

http://www.st-h.ed.jp/

21世紀の科学技術をリードする人材育成に主眼を置いたカリキュラムで生き生き学ぶ生徒

科学技術を専門的に学ぶ独自のカリキュラム

同校は，都立高校の改革第一号として誕生し，科学技術に関する基礎能力と，主に理工系大学への進学に対応したカリキュラムがしっかり組まれています。今年度からの学校5日制への対応として1日7時間授業（1単位時間45分）の実施や，休日となった土曜日を補習授業などにあてるなど，工夫も凝らされています。

科学や技術に関心が膨らむような実験・実習を重視した中にも，専門科目と普通科目との連帯が考えられているのも特徴です。たとえば，

- 「科学技術と人間」…科学と人間の係わりを数学・社会・工業などといった視点で考察し，実験を行う
- 「科学技術理論」…物理・生物・化学と関連を持たせ，力学・電磁気学・物質化学などの理論を学ぶ
- 「数理情報」…数学と関連させながらプログラミング実習を行う

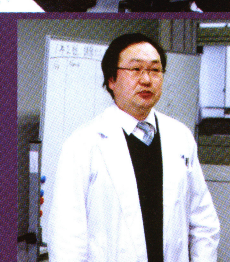
などがあります。

さらに，クラスは35名が定員ですが，各人の習熟度に応じた20名程度の少人数制授業を基本にし，16教室あるゼミ室（15名収容）での個別指導なども行われます。

1年次は全生徒が共通科目を学び，2年で興味・関心に応じた分野別制，3年で進路に合わせた学部別制となっています。その一方で，ゆとりをもたせた2学期制を採用しています。

就職のためさらに専門的に学びたいという生徒は，卒業後の2年間を「専攻科」（専門学校的なもの）で受け入れる体制も整っています。





情報技術室では生徒たちが率先して補習を受けていた。大坪靖幸先生のホームページ制作講座



▲「科学と技術は表裏一体をなす。日本の将来をたくす科学技術者を育成する第一段階の教育機関」と語る三上校長



▲今回の取材で案内していただいた市川政弘先生（MCG部顧問）



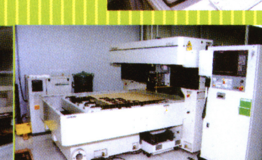
▲校長と並ぶ同校の“顔”と誰もが認める生徒会長の松村亜美子さん

実験棟の充実した設備



▶CAD実習室には1本300万円もするCADソフトが導入されている





◀家が一軒建つという高価なレーザー加工機

他校には例を見ない，充実した施設や設備

校舎は「設計の段階から意識した」（三上校長）という周りの景色と一体化したモダンな造りで，中庭にはペデストリアンデッキ（高架歩道）が教室棟と実験棟をむすんでいます。屋上にはソーラーパネルが設置されていて，ここで発電した電気は校舎内で使われています。

校舎正面には，大型のプラズマディスプレイがあり，学校予定など生徒への伝言板的な役目を持っています（操作は職員室のパソコンで）。

同校の設備の中でも大きな特徴は，4階建ての実習棟です。1Fはモノ作りに関する先端技術の実習を主にし，NC工作室やレーザー加工機などが設置されています。2Fはコンピュータ設計のCAD室が2教室のほか，高精度な計測器などが配置されています。3Fは化学が中心，基礎からバイオテクノロジーの実験などが行われます。4Fは情報部屋ともいうべき，すべての教室がパソコン室になっており，ネットワーク設計や，CG・ムービー編集などで使用されています。

中・高・大学・企業と連携した講義や集中講座も活発

今年の3月11日～13日，芝浦工業大学・水川研究室によるマイコン制御による『ライントレース・ロボットの製作』セミナーが行われました。同校では，春休みや夏休みなどの長期休暇前の数日間で「短期集中講座」が行われます。今回のロボット製作もその一環で行われたものです。

また，中学や大学・企業との連携による授業も活発で，大学の研究室や企業から技術者を招いた講義も頻繁に行われます。

逆に同校の先生や生徒が講師となって「子ども科学実験教室」なども開かれ，地域とともに科学技術の大切さを広く学習しています。

コンピュータ関連のクラブ活動も自発的に!!

MCG部

MCGとは，マルチメディア・コンピュータ・グラフィックスの略。現在1，2年生だけですが，約20人が，CG班，Flash班，ノベルゲーム班にわかれ，それぞれ興味のある事柄を集中的に学習しています。また各班連携によって作品作りにがんばっています。

具体的な活動は，学校説明会や中学配布用の学校紹介ビデオ制作，ポスター制作，Javaプログラミング，インターネット上でのフラッシュムービーやアニメーション制作，ゲーム作成などが主です。それぞれの作品は，各種コンテストや文化祭などへの出展も一つの目標にしており，情報デザイン室での活動が毎日続いています。

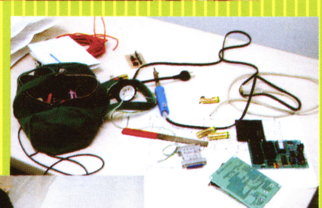














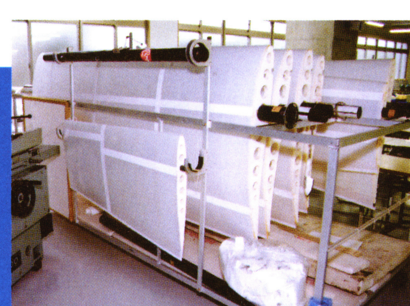
ロボット部

メカトロ好きな30名近くが在籍しています。現在は全国工業高等学校長協会などが主催する「マイコンカーラリー」（開催概要などはhttp://www.mcr.gr.jp）出場に向け，がんばっています。

部品は，東京・秋葉原の秋月電子通商などから購入，はんだ付けやプログラミング，試走の繰り返しの日々が続いているようです。



◀地域に開放されたり、作品展示スペースになったり、多目的に使用されるサイエンススクエア。奥に見える丸い形のものはプレネタリウム



◀鳥人間コンテスト出場の羽を発見！



東京都立科学技術高等学校 〒136-0072 東京都江東区大島1-2-31 ☎03-5609-0227



インテル春休みPCスクール

～パソコンで作ろう！「ぼくらの夢のまち」パソコンでオリジナル・ペーパークラフトを作ろう～

CPUメーカーのインテルが年に3回，小学校中・高学年生を対象に行っているPCスクールが，春休み中の3月28日に開催されました。

今年の春のテーマは「ぼくらの夢のまち」。3DCGソフト「六角大王」と展開図作成ソフトを使って立体模型を作り，みんなの夢の詰まった“街”を創ろうというもので，抽選で選ばれた都内近郊の4年生～6年生20人が参加しました。

午前中にパソコンやインターネットの基礎知識と，タイピング練習ソフトを使ってパソコンの操作を学んだ子どもたちは，インターネットを活用して夢のまちのイメージをふくらませました。

ランチタイムのあとは，いよいよ夢のまち作りの実践編。3Dソフト「六角大王」の使い方を学びながら，それぞれ思い思いの建物や，乗り物などを作成，展開図作成ソフトを通してプリントした作品を組み立て上げました。

完成した夢のまちは，子どもたちの手で，インテル東京本社のある国際ビル1階ロビーに展示されました。



▲インテル東京本社9階にあるトレーニングルームで授業スタート。今年度から4年生～6年生になる20人の子どもたち（男の子7人，女の子13人）が参加しました



▲マザーボードやCPUなど，ふだん見ることの少ないパーツを，実際に手に触れて“体験”する。覚えることより興味を持つことを優先するのがインテル流の学習法



▲毎回講師役を買って出る（他の人にはやらさせない？）というインテルの阿部剛士さん。手に持っているのはインテル自慢のシリコン・ウエハー（CPUの素材）



▲夢のまち作りを指導するのは，自称“ヨシザワ・キャップ”こと，講師の吉澤秀樹さん

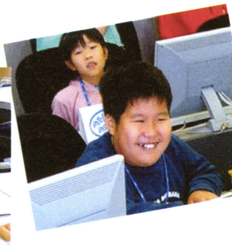


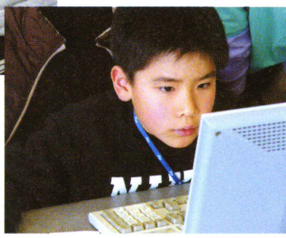
▲子ども向けワープロソフト「一太郎スマイル」のタイピング練習ソフト「キーボード・ファイター2」で，軽く肩ならし



▲四つのグループに分かれ，それぞれ“駅”，“公園”，“海”，“街”のエリア上に，まちを作ります。まずはエリアを決めるクジ引きから



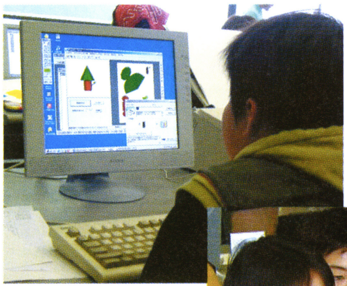




▲インターネットの検索サイトを駆使して，作るものをリサーチ







▲3DCGソフト「六角大王」で立体模型作り。初めて使うとは思えないくらい，みんな使いこなしていました





▲展開図作成ソフトをとおしてプリントアウト。あとは，ひたすら切って組み立てるべし



▲ぼくらの夢のまちの完成！

携帯電話で動くプログラムに挑戦!!

君にも作れる「iアプリ」



松原拓也

「iアプリ」とはNTTドコモのiモードに対応したJavaアプリケーションです。ゲームや実用的なソフトなど，さまざまなiアプリを携帯電話で楽しむことができます。

iアプリはユーザーが自分自身で作り出すこともできます。自作したiアプリを，携帯電話に入れて持ち歩いたり，サイトで公開して楽しめます。

今回はそんなiアプリの作成方法について特集します！

iアプリ対応機種

iアプリ対応機種「503i」シリーズは2001年1月の登場以来，契約者数はなんと1279万人にも達しています（2002年4月14日現在）。次世代携帯電話の「FOMA」や今後登場予定の「504i」でもiアプリは採用され，パソコンや家庭用ゲーム機に匹敵する巨大プラットホームとして成長しつつあります。

発売されている携帯電話の仕様（画面サイズや処理速度など）は，各社によって若干の違いがあります（**第1表**）。目的に合わせて選択するといいでしょう（ゲームを遊びたい場合はTFT液晶搭載機にするなど）。

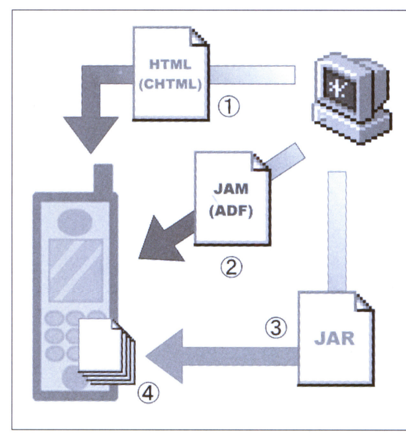
《第1表》 iアプリ対応機種一覧（*はFOMA端末）

型番	製造メーカー	重量	大きさ(*は折り畳み時)	液晶（カラー）	スクリーン・サイズ(ピクセル)
F503i	富士通	約77g	135×46×15mm	256色	120×130
P503i	松下通信工業	約74g	128×45×17mm	256色	120×130
So503i	ソニー	約115g	98×49×28mm*	65536色	120×120
N503i	日本電気	約98g	93×48×22mm*	4096色	120×130
D503i	三菱電機	約81g	129×46×17mm	4096色	132×126
F503iS	富士通	約97g	97×47×21mm*	4096色	120×130
P503iS	松下通信工業	約98g	97×50×27mm*	256色	120×130
So503iS	ソニー	約115g	98×49×27mm*	65536色	120×120
N503iS	日本電気	約105g	93×48×24mm*	4096色	120×130
D503iS	三菱電機	約105g	105×50×23mm*	4096色	132×126
N2002*	日本電気	約107g	103×52×22mm*	65536色	120×130
N2001*	日本電気	約105g	103×52×20mm*	4096色	120×130
P2101V*	松下通信工業	約150g	104×56×35mm*	262144色	176×182
D2101V*	三菱電機	約160g	150×55×28mm	262144色	132×162

iアプリの動作手順
（第1図の番号に連動）

①まず，iモードでWEBサイトを閲覧するところから始まります。HTMLファイル（CHTML）が携帯電話に読み込まれます。

②リンクを選択することで，携帯電話側にjamファイルが読み込まれます。jamファイルにはjarファイル（iアプリのプログラム本体）の所在が書かれています。



《第1図》 iアプリ実行までの流れ

③jarファイルをダウンロードします（128バイトあたり0.3円のパケット通信費がかかります）。jarファイルは携帯電話のメモリに記録されますので，ネットワークに接続しないタイプなら，それ以上通信費はかかりません。ダウンロードできるプログラムサイズは10kバイトまでに制限されています（今後，このサイズの上限を拡大する計画のようです）。

④jarファイルはzipで圧縮されています。実行時にはRAM上に一旦解凍してから実行されます。

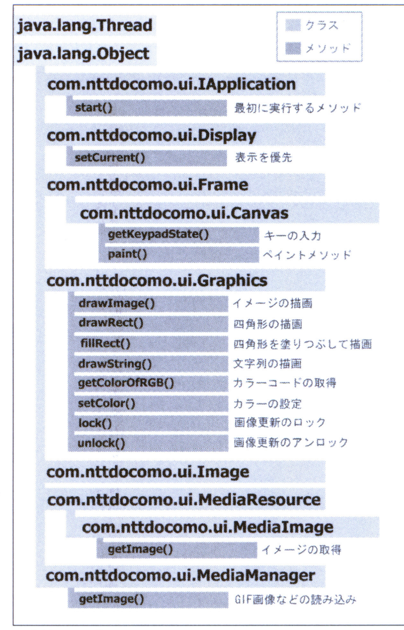
com.nttdocomoクラス

iアプリのプログラミング言語はJ2ME CLDC（Java 2 Micro Edition, Connected Limited Device Configuration）

という携帯通信端末用に設計されたJavaがベースになっています。

基本的な制御は“com.nttdocomo”というｉアプリ専用のクラスを使います(**第2図**)。クラスの仕様書（pdf）はNTTドコモのWEBサイトからダウンロードすることができます。

ｉアプリの画面制御は「Panel」と「Canvas」という2種類のクラスのどちらかを使って行います。「Panel」はテキストボックスなどのオブジェクトを扱う場合に適していて，高レベルAPIと呼ばれています。一方「Canvas」は直接画面の制御を行う場合に適しています。

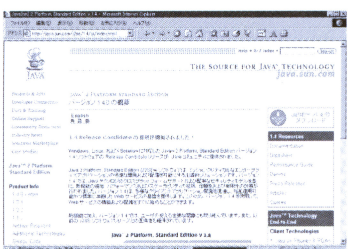


《第2図》com.nttdocomoクラス

ｉアプリの開発環境を作ろう

ｉアプリの開発には「Java2 SDK」「クラスライブラリ」「携帯電話エミュレータ」が必要です。これらのツールはすべてインターネット上から無償で手に入れることができます。

『Java2 SDKの入手』



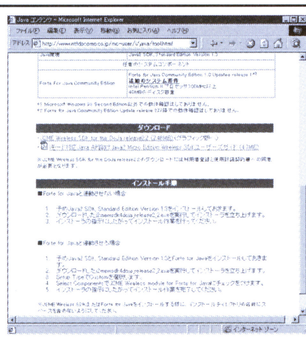
サンマイクロシステムズのWEBサイト（http://java.sun.com/）からJavaの開発環境「Java 2 SDK Standard Edition（以下，JDK）」をダウンロードします。JDKの最新バージョンは1.4ですが，1.3以上ならどのバージョンでも問題なくｉアプリ開発に使えます。

インストールすると指定のディレクトリにJava用のコンパイラとJavaVM（virtual machine）が展開されます。

「Java 2 Platform, Standard Edition バージョン 1.4」
http://java.sun.com/j2se/1.4/ja/index.html

「Java 2 Platform, Standard Edition バージョン 1.3.1-02」
http://java.sun.com/j2se/1.3/ja/index.html

『ライブラリとエミュレータの入手』

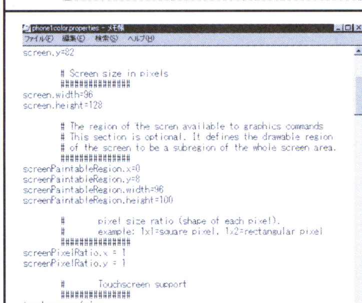


NTTドコモのWEBサイト（http://www.nttdocomo.co.jp/）から「J2ME Wireless SDK for the DoJa release2.2（以下，DoJa）」をダウンロードします。「DoJa」はｉアプリ用クラスライブラリと携帯電話エミュレータを含んだ，ｉアプリの開発ツールです（対応機種はWindows98）。ダウンロードするためには名前とメールアドレスを入力する必要があります。

インストールが終わると主要なファイルが“C:¥J2MEWSDK4DoJa”フォルダに展開されます。

NTTドコモ「作ろうｉモードコンテンツ　Java」
http://www.nttdocomo.co.jp/mc-user/i/java/index.html

『エミュレータの設定』



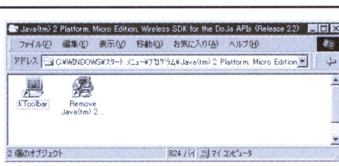
▲ｉアプリ用にエミュレータを設定する

エミュレータは，携帯電話がなくてもパソコン上でｉアプリを実行することができるツールです。付属のエミュレータにはｉアプリ用を実行するための設定がありません。そこで，エディタ（またはメモ帳）を使って“C:¥J2MEWSDK4DoJa¥lib¥devices¥phone1color”フォルダにある“phone1color.properties”というファイルを書き換えます。これによって，画面のサイズやアイコン・ボタンの位置を修正します。

▼F503i/P503i/N503iの画面サイズ（120×130ピクセル）の例

```
screen.x=25
screen.y=65
（中略）
screen.width=120
screen.height=156
（中略）
screenPaintableRegion.width=120
screenPaintableRegion.height=130
（中略）
icon.up: 44, 138, off
（中略）
icon.down: 44, 146, off
（中略）
softbutton.0=0,138,36,18, softButton
softbutton.1=90,138,36,18, softButton
```

『DoJa』の実行



▲スタートメニューに「KToolBar」が登録される

スタートメニューから「Java(tm) 2 Platform, Micro Edition, Wireless SDK for the DoJa APIs (Release 2.2)」を選択して，「KToolBar」を実行します。 そして「編集ー設定」をメニュー選択して，「デバイス指定」を“phone1color”にします。

これでｉアプリの動作環境は整いました！

iアプリ作成方法

では，手始めに画面に「Helloベーマガ」と表示させるｉアプリを作ってみましょう。これは文字列が表示されるだけの最も簡単なプログラムです。

『プロジェクトの作成』

スタートメニューから「J2ME～DoJa」を選択して，「KToolBar」を実行します。

ツールバーの「新規」ボタンを押して，さらに新しく作りたいプロジェクト名を入力します（**写真1**）。入力すると，“C:¥J2MEWSDK4DoJa¥apps”フォルダの下にプロジェクト名のフォルダと“bin”，“res”，“src”というフォルダが作成されます（**写真2**）。プロジェクト名を“hello”とすると，フォルダやファイルの名前も“hello”になります（「hello.jam」「hello.jar」）。

『ソースの作成』

“src”フォルダにソースファイルを作成します。

エディタ（または「メモ帳」）を使い，**リスト1～2**のように“TestApp.java”と“TestScreen.java”という二つのファイルを作ります。拡張子を“java”にして保存しましょう（**リスト1，リスト2の番号に連動**）。

①ｉアプリは「IApplication」クラスから派生させたクラスから始まります。このクラスがメインプログラムになります。

②「IApplication」の派生クラスでは「start」メソッドが最初に1度実行されます。

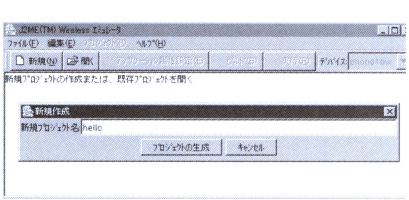
③Canvasから派生させた「myScreen」オブジェクトを生成します。

④「Display.setCurrent(myScreen);」を実行することでオブジェクトが画面に表示されます。

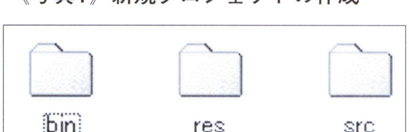
⑤「Canvas」クラスから「TestScreen」クラスを派生させます。このクラスがキー入力と画面更新の処理を受け持ちます。

⑥オブジェクト定義時に最初に1度実行されるメソッドを「コンストラクタ」といいます。「setSoftLabel」関数でソフトキー2の文字を設定しています。

⑦「paint」メソッドは画面更新時に呼び出されます。「lock」で画面更新を固定します。「fillRect」で四角形の描画，そして「drawString」で文字を表示させます。「unlock」で開放して始めて画面が更新されます。



《写真1》新規プロジェクトの作成



《写真2》自動的に作られた3つのフォルダ

《リスト1》“TestApp.java”

```
/*-------------------------------------------------------------*/
/* "TestApp.java"  */

import com.nttdocomo.util.*;  /*クラスのインポート*/
import com.nttdocomo.ui.*;

public class TestApp extends IApplication ◀……………………………①
{
   public void start() {    ◀……………………………………………②
        TestScreen screen = new TestScreen(); /*キャンバスの作成*/◀………③
           Display.setCurrent(screen);  /*キャンバスのカレント表示*/◀……④
   }
}
```

《リスト2》“TestScreen.java”

```
/*-------------------------------------------------------------*/
/* キャンバス "TestScreen.java"  */

/* クラスをインポート*/
import com.nttdocomo.ui.*;
import com.nttdocomo.io.*;

public class TestScreen extends Canvas { ◀…………………………………⑤

   public TestScreen() {   /*コンストラクタ*/◀…………………………⑥
      setSoftLabel(Frame.SOFT_KEY_2, "終了");
   }

   public void paint(Graphics g) {  /*ペイントメソッド*/◀………………⑦
      g.lock();  /* 画面のロック */
      g.setColor(g.getColorOfRGB(0, 0, 255));
      g.fillRect(0, 0, getWidth(), getHeight() );
      g.setColor(g.getColorOfRGB(255, 0, 0));
      g.drawString("Helloベーマガ", 20 ,50);
      g.unlock(false);  /* ロック解除 */
   }

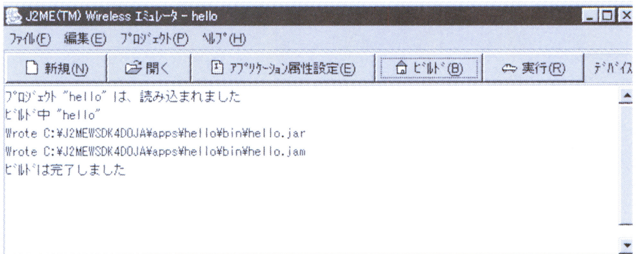
   public void processEvent(int iType, int iParam) {  ◀…………………⑧
      if (iType == Display.KEY_PRESSED_EVENT) { /* キープレスイベント */
         if(iParam == Display.KEY_SOFT2) {  /* ソフトキー2が押された */
            IApplication.getCurrentApp().terminate(); /* アプリ終了 */
         }
      }
   }
}
```

『その他の開発ツール』

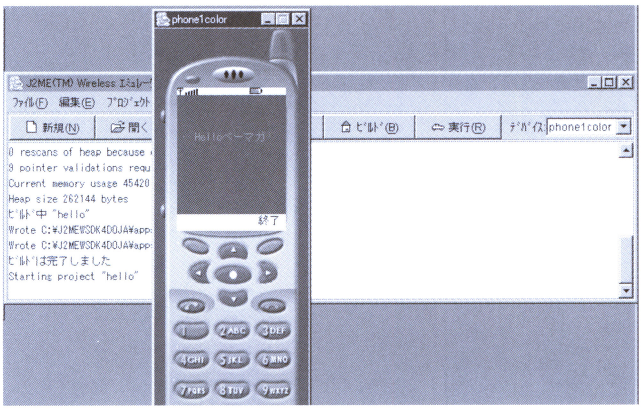
・『Forte for Java』（サンマイクロシステムズ）
http://www.sun.co.jp/forte/ffj/
Javaの総合開発環境です。「Community Edition」は無償で配布されています。

・『JBuilder4 Foundation』（ボーランド）
http://www.borland.co.jp/jbuilder/foundation/
無償で配布されているJavaの開発ツールです。

・『i-JADE Lite』（ゼンテック・テクノロジー・ジャパン）
http://www.zentek.com/i-JADE/
無償で配布されている携帯電話エミュレータです。



《写真3》ビルドに成功した様子



《写真4》エミュレータでの実行結果。「Helloベーマガ」と表示された

《リスト3》 **"hello.jam"**

```
PackageURL = hello.jar
AppSize = 1061
AppName = Helloベーマガ
AppVer =
AppParam =
AppClass = TestApp
SPsize = 0
LastModified = Sat, 16 Feb 2002 07:02:30
KvmVer =
```

《リスト4》 **"hello.html"**

```
<HTML>
<HEAD>
<TITLE>hello</TITLE>
</HEAD>
<BODY>
「Helloベーマガ」<BR>
<OBJECT declare id="b" data="hello.jam" type="application/x-
jam"></OBJECT>
<A ijam="#b" href="hello.html">DOWNLOAD </A><BR>
「Helloベーマガ」と表示するだけのｉアプリです。<BR>
</BODY>
</HTML>
```

⑧「processEvent」メソッドでキープレスイベント（Display.KEY_PRESSED_EVENT）を受け付けます。ソフトキー2が押された場合，「terminate」メソッドでアプリケーションを終了させます。

『アプリケーション属性の設定』

「アプリケーション属性設定」ボタンを押して，値を入力します。“AppName”はアプリケーション名，“AppClass”には最初に実行するクラス名を入力します。設定データは**リスト3**のようなjamファイルに反映されます。

「AppName」：メニュー登録時の名前を定義します。日本語でもかまいません。

「AppClass」：最初に実行するクラス名を定義します。

「PackageURL」：読み込むjarファイル名を定義します。

「SPsize」：スクラッチパッドのサイズを定義します。未使用の場合は0です。

「AppSize」：jarのファイルサイズを1バイト単位で定義します（自動的に更新されます）。

「LastModified」：ファイルの更新日を定義します（自動的に更新されます）。

『ビルド』

ツールバーの「ビルド」ボタンを押すと，コンパイル（javac）から圧縮（jar）までの処理が一括して行われます。ビルドに成功すると“bin”フォルダにjarファイルとjamファイルが生成されます（**写真3**）。

『エミュレータの実行』

ツールバーの「実行」を押すとエミュレータを使ってパソコン上でｉアプリが実行されます。画面に「Helloベーマガ」と表示されました!!（**写真4**）

ソフトキー2を押すとプログラムを終了します。

『アップロード』

今度はエミュレータではなく，実機でｉアプリを動かしてみましょう。エディタを使い，**リスト4**のようなhtmlファイルを作ります。

あとはFTPクライアントソフトを使って，jarファイルとjamファイル，htmlファイルをサイトにアップロードします。保存先は自分のホームページ上でかまいません。

『実行』

携帯電話のｉモードを使って，アップロードしたサイトを閲覧します。参照するURLは……

http://（保存先のURL）/hello.html

……となります。ｉアプリへのリンクを選択してボタンを押すと，ｉアプリが携帯電話にダウンロードされます。携帯電話のメニューからｉアプリを実行すると見事に動きました!! オリジナルｉアプリの完成です（**写真5**）。



《写真5》実機での実行結果！

サウンド再生の方法

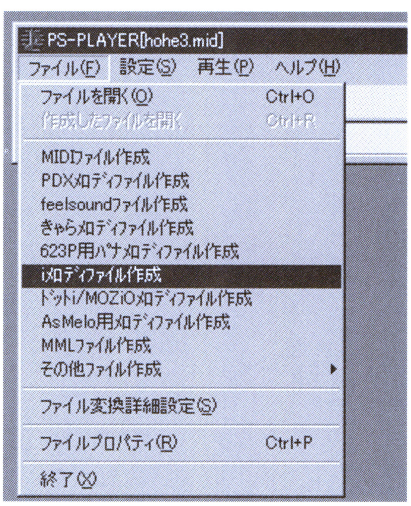
ｉアプリは「ｉメロディ」という形式のサウンドを再生できます（1～4トラック4和音，503iは4トラック16和音）。

そこで，MIDIのデータをｉアプリで鳴らしてみたいと思います。まず各種シーケンスソフトや楽器を使ってMIDIの曲を作成します（拡張子は“mid”）。そして，フリーウェアの「PS-PLAYER」を使って，MIDIからｉメロディへデータを変換します。

まず「PS-PLAYER」（ps-player.exe）を立ち上げてMIDIファイルを読み込みます。そして，「ファイルーｉメロディファイル作成」をメニュー選択すれば，データがｉメロディ（拡張子は“mld”）ファイルになって保存されます（**写真6**）。

サンプルプログラム

リスト5がサウンドを再生するサンプルです。ｉアプリを実行して，上のキーを押すとサウンドの再生。下を押すとサウンドの再生を停止します（**写真7**）。



《**写真6**》「**PS-PLAYER**」で**i**メロディに変換（対応機種は**Windows95/98/Me/XP**）



《**写真7**》実行画面。サウンドを再生中

プログラムの解説

（**リスト5**の番号に連動）

①サウンド再生用のオブジェクト「AudioPresenter」を定義します。

②「MediaManager」オブジェクトの「getSound」メソッドでmldファイルを読み込みます。音楽データは「MediaSound」に読み込まれます。

③「setSound」メソッドで音楽データを「MediaSound」から「AudioPresenter」オブジェクトに転送します（この時点で「MediaSound」オブジェクトは破棄できます）。

④「play」メソッドでメロディが再生されます。停止は「stop」メソッドです。

「ＰＨ″Ｓ SOFT HOUSE」
（「PS-PLAYER」の作者，APOさんのWEBサイトです）
http://www.ne.jp/asahi/phs/phs

＊注意　「DoJa」付属のエミュレータではmldファイルを再生することができません（エラーになります）。このためデバッグ中は代わりにMIDIファイル（mid）を鳴らし，実機で動作させる前にmldに切り替えます。なお，「i-JADE」ではきちんとmldファイルが再生されます。

《リスト5》**“TestScreen.java”**

```
/*-----------------------------------------------------------------*/
                    /* キャンバス "TestScreen.java" */

/* クラスをインポート*/
import com.nttdocomo.ui.*;
import com.nttdocomo.io.*;

public class TestScreen extends Canvas {
  AudioPresenter audio; ◀·························· ①

  public TestScreen() {   /*コンストラクタ*/
    MediaSound ms;

    setSoftLabel(Frame.SOFT_KEY_2, "終了");
    /* i メロディ読込*/
    ms = MediaManager.getSound("resource:///music.mld"); ◀·········· ②
    try{
      ms.use();
    }catch (ConnectionException e){
    }
    audio = AudioPresenter.getAudioPresenter();
    audio.setSound(ms); ◀·························· ③
  }

  public void paint(Graphics g) { /*ペイントメソッド*/
    g.lock(); /* 画面のロック */
    g.setColor(g.getColorOfRGB(0, 0, 0));
    g.fillRect(0, 0, getWidth(), getHeight() );
    g.setColor(g.getColorOfRGB(255, 255, 255));
    g.drawString("[↑]:再生 [↓]:停止" , 0 ,60);
    g.unlock(false); /* ロック解除 */
  }

  public void processEvent(int iType, int iParam) {
    if (iType ==Display.KEY_PRESSED_EVENT) { /* キープレスイベント */
      switch(iParam) {
      case Display.KEY_SOFT2: /* ソフトキー2 */
        IApplication.getCurrentApp().terminate(); /* アプリ終了 */
        break;
      case Display.KEY_UP: /* ↑キー */
        audio.play(); ◀·························· ④
        break;
      case Display.KEY_DOWN: /* ↓キー */
        audio.stop();
        break;
      }
    }
  }
}
```

スクラッチパッドの使い方

ｉアプリには「スクラッチパッド(ScratchPad)」という，携帯電話内にデータを保存させる機能があります。

スクラッチパッドのサイズはアプリケーション設定（jamファイル）の“SPsize”キーで指定します（**写真8**）。“SPsize”の値が0だと保存されませんので注意しましょう。現行の携帯電話（503i）はトータルで10kバイトほどのスクラッチパッドを搭載しています。

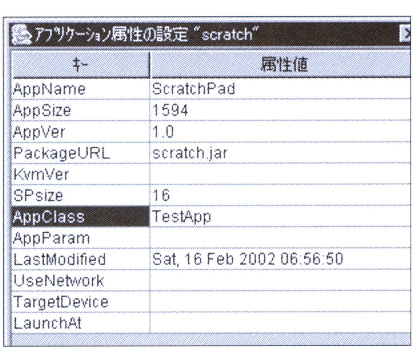
サンプルプログラム

リスト6がサンプルプログラムです。上下キーで値を変えて，それからソフトキー2でいったんアプリを終了させせます。そして，再び起動するとスクラッチパッドに書かれた値を読み出します（**写真9**）。

プログラムの解説

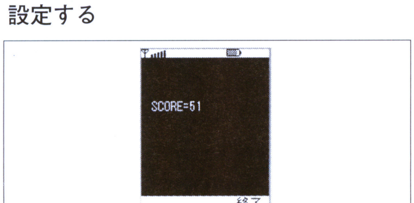
① “openDataInputStream”メソッドでオブジェクトを取得します。“readInt”でスクラッチパッドの読み込みを行います（サイズ別に“readByte”や“readLong”があります）。

② “openDataOutputStream”メソッドでオブジェクトを取得します。“writeInt”でスクラッチパッドの書き込みを行います（サイズ別に“writeByte”や“writeLong”があります）。


アプリケーション属性の設定 "scratch"

キー	属性値
AppName	ScratchPad
AppSize	1594
AppVer	1.0
PackageURL	scratch.jar
KvmVer	
SPsize	16
AppClass	TestApp
AppParam	
LastModified	Sat, 16 Feb 2002 06:56:50
UseNetwork	
TargetDevice	
LaunchAt	

《写真8》スクラッチパッドの大きさを設定する



《写真9》実行画面。アプリケーションを終了しても値は残り続ける！

《リスト6》“TestScreen.java”

```
/*---------------------------------------------------------------*/
/* キャンバス "TestScreen.java"   */

/* クラスをインポート*/
import com.nttdocomo.ui.*;
import com.nttdocomo.io.*;
import javax.microedition.io.*;
import java.util.*;
import java.io.*;

public class TestScreen extends Canvas implements Runnable {
   Thread myThread; /* スレッド */
   int iScore=0;   /*スコア*/

   public void readSP() {   /*スクラッチパッド読み込み*/
      try {
         DataInputStream in=Connector.openDataInputStream("scratchpad:///0"); ◀……… ①
         iScore = in.readInt();
         in.close();
      }
      catch(IOException e) {
         /* 例外処理 */
      }
   }

   public  void writeSP() {   /*スクラッチパッド書き込み*/
      try {
         DataOutputStream out=Connector.openDataOutputStream("scratchpad:///0"); ◀……… ②
         out.writeInt(iScore);
         out.close();
      }
      catch(IOException e) {
         /* 例外処理 */
      }
   }

   public TestScreen() {   /*コンストラクタ*/
      setSoftLabel(Frame.SOFT_KEY_2, "終了");

      myThread = new Thread(this); /* スレッド作成 */
      myThread.start(); /* スレッド開始 */

      readSP();   /* スクラッチパッド読み込み */
   }

   public void paint(Graphics g) { /*ペイントメソッド*/
      g.lock(); /* 画面のロック */

      g.setColor(g.getColorOfRGB(0, 0, 0));
      g.fillRect(0, 0, getWidth(), getHeight() ); /*黒*/
      g.setColor(g.getColorOfRGB(255, 255, 255));
      g.drawString("SCORE="+iScore, 10, 50);

      g.unlock(false); /* ロック解除 */
   }

   public void run() { /* スレッド実行 */
      int iKey;

      for(;;) {
         try {
            Thread.sleep(100);
         } catch(InterruptedException ignore) { }

         iKey=getKeypadState(); /* キー入力 */

         if(iKey==(1 << Display.KEY_UP))
             iScore++;

         if(iKey==(1 << Display.KEY_DOWN))
             iScore--;

         if(iKey==(1 << Display.KEY_SOFT2)) { /*ソフトキー2*/
            writeSP();      /* スクラッチパッド書き込み */
            IApplication.getCurrentApp().terminate(); /*終了*/
         }
         repaint(); /*再描画*/
      }
   }
}
```

ゲームの作り方

「おやつのホームラン王」というゲームを作成してみました。遊び方ですが，「おやつ（ういろう？）」が落下してきますので，「i（選択）」キーをタイミングよく押して叩き切ってください。ここで中心を切るほど高得点になります。おやつの落下速度は徐々に上がっていきます。おやつが地面に落ちるとゲームオーバーです（**写真10，11**）。



《**写真10**》ゲーム「おやつのホームラン王」



《**写真11**》切り損ねるとゲームオーバー

入力方法

リスト7～9を入力して，“src”フォルダ内に保存します。そして，“my1.gif”と“my2.gif”の2枚の画像（gif）ファイルをペイントソフトを使って作成して“res”フォルダに保存します（**写真12**）。ビルドは前述した方法で行ってください。

プログラム解説

（リスト7の番号に連動）

① “Random”クラスで乱数を作成しま

《リスト7》“GameScreen.java”

```
/*---------------------------------------------------------------*/
/*  キャンバス "GameScreen.java"   by 松原拓也 */

/* クラスをインポート*/
import com.nttdocomo.ui.*;
import com.nttdocomo.io.*;
import java.util.*;

public class GameScreen extends Canvas implements Runnable {
    final int MY_W = 32;            /*主人公幅*/
    final int MY_H = 64;            /*主人公高さ*/
    final int EM_W = 8;             /*おやつの幅*/
    final int EM_H = 40;            /*おやつの高さ*/
    final int SEQREADY = 0;         /*シーケンス：レディ*/
    final int SEQGO = 30;           /*シーケンス：GO*/
    final int SEQSTART = 40;        /*シーケンス：プレイ中*/
    final int SEQWIN = 41;          /*シーケンス：おやつの吹っ飛び*/
    final int SEQGAMEOVER = 42;     /*シーケンス：ゲームオーバー*/
    final int PUNCHMAX = 14;        /*パンチ中*/
    final int PUNCHMIN = -PUNCHMAX; /*パンチ待機中*/
    final int GAME_SPEED = 100;     /*ゲームのスピード*/

    Thread myThread;  /* スレッド */
    Image imgMy1;     /* イメージ:主人公（通常状態） */
    Image imgMy2;     /* イメージ:主人公（突き状態） */
    int iMx,iMy,iPunch; /* 主人公座標 */
    int iEx,iEy1,iEy,iEa,iEh; /* 敵座標 */
    int iSeq;    /*シーケンス*/
    int iScore;     /*スコア*/
    int iTotal;     /*トータルスコア*/
    int iNando;     /* 難易度 */
    Random rnd = new Random();  /* 新しい乱数列を作成*/ ◀……………… ①

    public Image getImg(String sName) { /* イメージ読み込み関数 */
        MediaImage mi = MediaManager.getImage(sName); ◀……………… ②
        try {
            mi.use(); /* イメージの読み込み */
        } catch (ConnectionException e) {
        }
        return(mi.getImage());
    }

    public void subInit() {   /*ゲームデータのクリア*/
        iMx = 20;  /* 主人公座標 */
        iMy = 45;
        iEx = 80;  /* おやつの出現座標 */
        iEy = 0;
        iEy1 = 0;
        iPunch = PUNCHMIN;
        iSeq = SEQREADY;  /*シーケンス初期化*/
        iTotal = 0;         /*スコア0点*/
        iNando = 97;        /*難易度2*/
    }

    public GameScreen() {   /*コンストラクタ*/
        setSoftLabel(Frame.SOFT_KEY_2, "終了");

        imgMy1 = getImg("resource:///my1.gif"); /*GIF画像読込*/
        imgMy2 = getImg("resource:///my2.gif");

        myThread = new Thread(this); /* スレッド作成 */ ◀……………… ③
        myThread.start();  /* スレッド開始 */

        subInit();  /*ゲームデータのクリア*/
    }

    public void paint(Graphics g) { /*ペイントメソッド*/
        g.lock();  /* 画面のロック */

        g.setColor(g.getColorOfRGB(255, 255, 255)); /*画面クリア*/
        g.fillRect(0, 0, getWidth(), getHeight() );

        g.setColor(g.getColorOfRGB(80, 80, 80)); /*地面*/
        g.fillRect(0, (45+64), 120, (120-(45+64)) );

        g.setColor(g.getColorOfRGB(255, 255, 255));  /* 主人公の描画*/
        if(iPunch == 0) {
            g.drawImage(imgMy2 ,iMx ,iMy);
            g.setColor(g.getColorOfRGB(0, 128, 255));
            g.fillRect(iMx+MY_W ,iMy+(MY_H/2) ,30 ,4 );  ◀……………… ④
        }else{
            g.drawImage(imgMy1 ,iMx ,iMy);
        }

        g.setColor(g.getColorOfRGB(255, 0 ,128)); /*おやつの描画*/
        if(iSeq == SEQWIN) {
            g.fillRect(iEx+iEa, iEy-iEa, EM_W, iEh ); /*真っ二つ*/
            g.fillRect(iEx+iEa, (iEy+iEh)+iEa, EM_W, (EM_H-iEh) );
        }else{
            g.fillRect(iEx, iEy, EM_W, EM_H );
        }
        g.setColor(g.getColorOfRGB(0, 0 ,0));
        g.drawString("Total Score="+iTotal, 0, 12);

        if(iSeq < SEQGO) {  /*シーケンス別のメッセージ*/
            g.drawString("READY...", 40, 40);
```

→ つぎのページにつづく

```
        }else if(iSeq < SEQSTART) {
            g.drawString("...GO!!", 40, 40);
        }else if(iSeq == SEQWIN){
            g.drawString("Score "+iScore, 40, 40);
        }else if(iSeq == SEQGAMEOVER){
            g.drawString("GAME OVER!!", 40, 40);
        }
        g.unlock(false);  /＊ ロック解除 ＊/
    }

    public void run() { /＊ スレッド実行 ＊/
        int iKey;

        for(;;) { ◀‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥ ⑤
            try {
                Thread.sleep(GAME_SPEED);
            } catch(InterruptedException ignore) { }

            iKey=getKeypadState();      /＊ 主人公 ＊/

            if(iSeq < SEQSTART) {
                iSeq++;     /＊READY...GO!!シーケンス＊/
            }else if(iSeq == SEQWIN){
                iEa +=2;    /＊お菓子、吹っ飛び中＊/
                if(iEa >= 100) {
                    iEy = 0;
                    iEy1 = 0;   /＊初期化＊/
                    iTotal += iScore;
                    iNando--;  /＊難易度プラス＊/
                    iSeq = SEQSTART;
                }
            }else if(iSeq == SEQGAMEOVER){
                iMx -= 1;  /＊GAMEOVER中＊/
                iMy -= 2;
                if(iMx < (-MY_W) ) {
                    subInit();  /＊ゲームデータのクリア＊/
                }
            }else{
                iMx= 20+(PUNCHMAX - Math.abs(iPunch));

                if(iPunch > PUNCHMIN ) { /＊パンチ状態チェック＊/
                    iPunch – ;     /＊パンチ＊/
                }else{
                    if(iKey==(1 << Display.KEY_SELECT))
                        iPunch=PUNCHMAX;    /＊パンチ開始＊/
                }

                if(iPunch == 0) {   /＊当り判定＊/
                    iEh =  (iMy+(MY_H/2))-iEy;
                    if( (iEh > 0) && (iEh < EM_H)) {
                        iScore = (EM_H/2)-Math.abs(iEh - (EM_H/2));
                        iEa=0; /＊お菓子吹っ飛び量=0＊/
                        iSeq = SEQWIN;
                    }
                }

                if( Math.abs(rnd.nextInt() % 100) > iNando)
                    iEy1 ++;  /＊落下の加速＊/

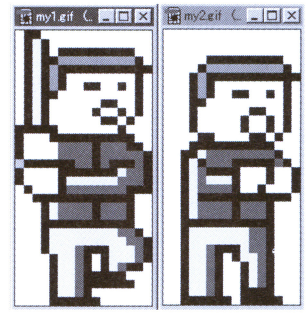
                iEy += iEy1;  /＊おやつの落下＊/
                if(iEy > 120)
                    iSeq=SEQGAMEOVER; /＊地面に付くとGAMEOVER＊/
            }

            if(iKey==(1 << Display.KEY_SOFT2)) {
                IApplication.getCurrentApp().terminate(); /＊終了＊/
            }
            repaint(); /＊再描画＊/◀‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥‥ ⑥
        }
    }
}
```

す。"Random"クラスを使うためには"java.util.＊"をインポートしておきましょう。

② "MediaManager"の"getImage"メソッドでgif画像を読み込みます。読み込んだ画像データは"Image"に格納されます。

③スレッドを作成して実行します。これによってリアルタイム性のある処理が行えます。



《写真12》作成する2枚のgif画像(32×64pixel)

④ "fillRect"メソッドで，プレイヤーのキャラクターが剣を突き出している時の「剣の部分」を描きます。

⑤スレッド内でメインの処理を実行させます（無限ループ)。

⑥ "paint"メソッドを呼び出して，画面を再描画させます。

最後に

ｉアプリは，ソースコードの短さや，開発ツールがほとんどの無料で手に入れられるという手軽さから考えて，Javaプログラミングの入門に向いていると思います（本誌への投稿用としても)。

あえて，欠点を挙げるとすれば，割高なパケット代と携帯電話の非力さが問題でしょうか。今年春登場予定の「504i」では回線の高速化やｉアプリの大容量化(30kバイト）が予定されています。通信料金の変化にも期待しています。

今回解説したサンプルプログラムはマイコンBASIC MagazineのWEBサイトからダウンロードすることができます。
http://www.basicmagazine.net/

《リスト8》 "GameApp.java"

```
/＊----------------------------------------------------------------＊/
/＊ ゲーム"GameApp.java"  by 松原拓也＊/

import com.nttdocomo.util.＊;  /＊クラスのインポート＊/
import com.nttdocomo.ui.＊;

public class GameApp extends IApplication {
    public void start() {
        GameScreen screen = new GameScreen(); /＊キャンバスの作成＊/
        Display.setCurrent(screen);  /＊キャンバスのカレント表示＊/
    }
}
```

《リスト9》 "Homerun.html"

```
PackageURL = homerun.jar
AppSize = 3599
AppName = ホームラン王
AppVer =
AppParam =
AppClass = GameApp
SPsize = 0
LastModified = Fri, 15 Feb 2002 02:05:57
KvmVer =
```

見やすい，楽しい，誰でも使えるサイトを実現！

一歩先ゆくホームページ作り

第3回 リンク…ページとページをつなげる，よそのホームページとつなげる

白木ミホ

本誌読者のキミなら，作成支援ソフトを使って，なんとなくできちゃったなんてホームページではなく，タグの基本をしっかり理解し，マシンやブラウザ，通信回線環境などに左右されないユニバーサルデザインのホームページをめざそう！　今回はＨＰを構築し，さらにインターネットの世界に結びつけるための「リンク」を極めます。

リンクは総称

日進月歩のインターネット世界では，ことがらに名前が追いつかないこともたくさんあります。その一つが「ハイパーリンク」（通称リンク）です。ひとくちにリンクといっても，実はいろいろな種類があるのです。

一つは，複数のhtmlファイルをつなげ，一つのホームページとして完成させる「自サイト内リンク」で，ページとページをつなぐリンクです。

もう一つは，自分のホームページと他の人のホームページをつなげる「他サイト間リンク」です。自サイト内リンクはあくまでもキミひとりの作業ですが，他サイト間リンクでは，他のホームページの管理人さんとの関係が出てきます。

さらに，一般的なリンクがリンク元のページからリンク先のページのいちばん上（トップ）へ「飛んで」行くのに対し，トップ以外の指定の場所へ飛べるようにするリンクもあります。これも，同じページ内の指定の場所へリンクする方法と，他ページの指定の場所へリンクする方法の二つがあります。（第1図）

また，ページでなく，メールソフトが立ちあがるリンクもあります。

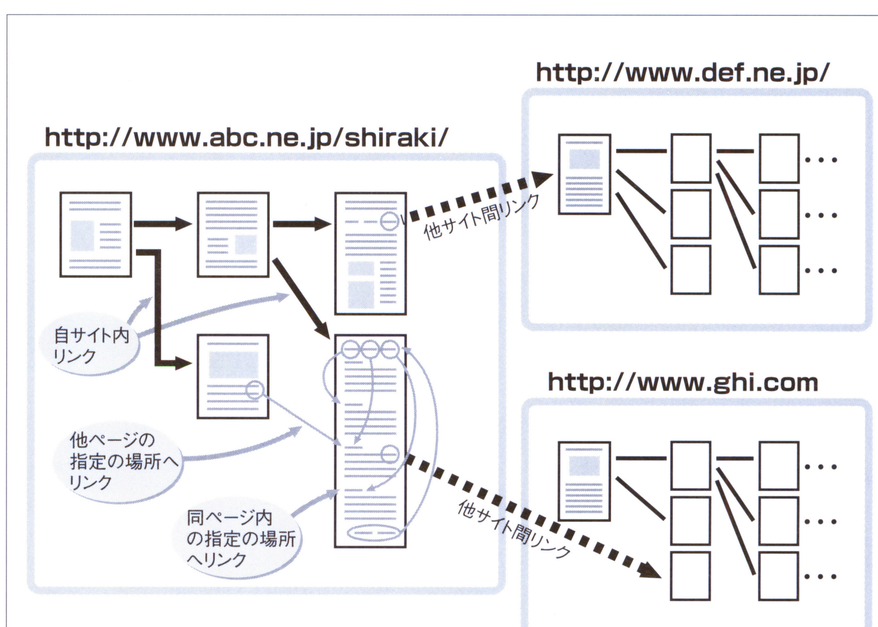
このように，ひとくちにリンクといっても機能は実にさまざまなのです。

絶対パスと相対パス

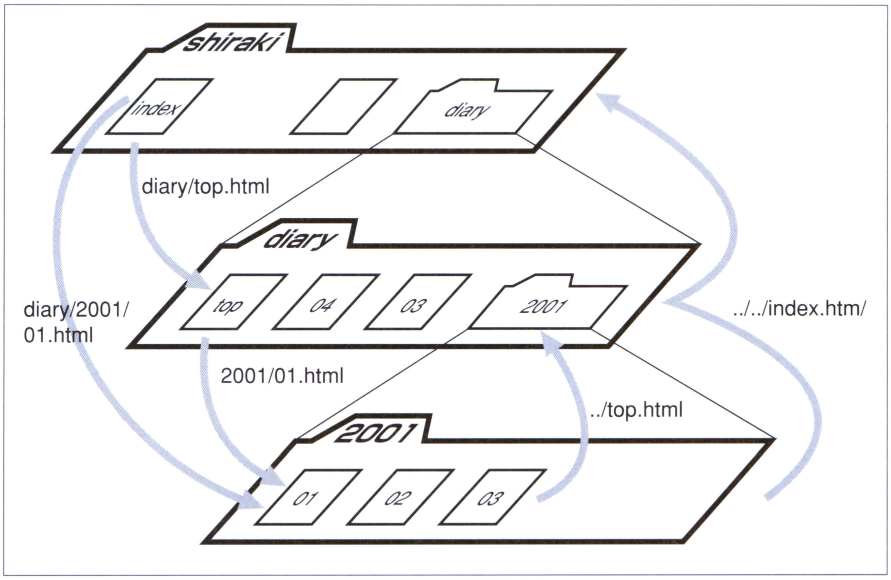
リンクを表すタグはと書きます。もちろん，タグの常識，タグもあります。

"　"の中にリンク先のアドレスを書き込むのですが，この書き方には「絶対パス」と「相対パス」の２とおりがあります（今回，「２とおり」だらけですが，なんとかこらえてやってください）。

一つは「絶対パス」と呼び，「http://」から始まるアドレスのすべてを書き込む



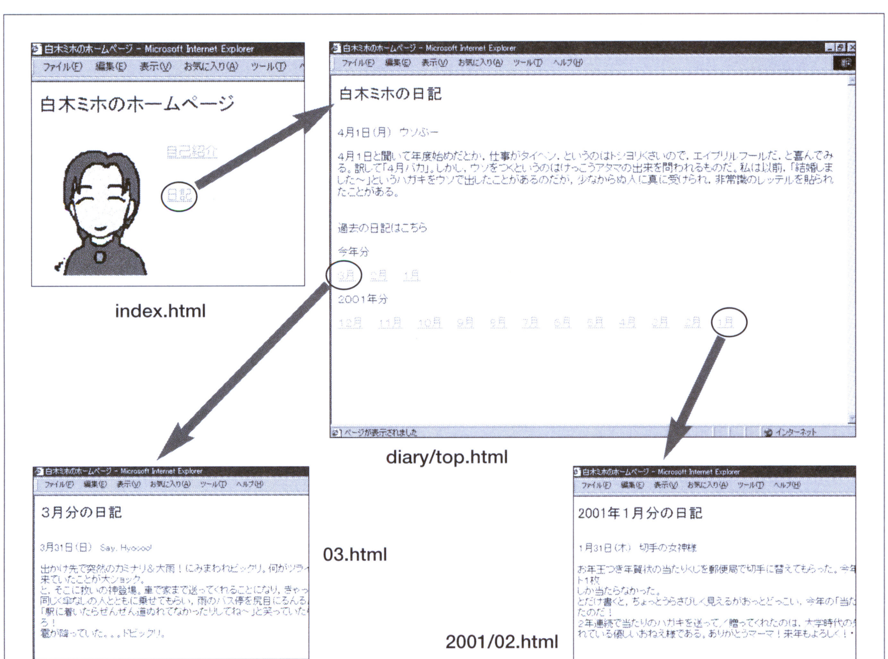
《第１図》ファイルのリンクいろいろ。こうして張りめぐらされたリンクがネットサーフィンを可能にするのです



《第２図》階層の上下が決める相互リンクの形。「/」（スラッシュ）でフォルダ名が区切られています

《リスト1》第3図の画面のソースリスト

```
<HTML>
<HEAD>
<TITLE>白木ミホのホームページ</TITLE>
</HEAD>
<BODY bgcolor="#ffffff" text="#000000" link="#ff0000" vlink="#000099" alink="#0000ff">
<P>
<FONT size="+2">白木ミホのホームページ</FONT><BR>
<BR>
<BR>
<IMG src="shiraki.gif" alt="百万ドルの笑顔" hspace="10" align="left"><A
href="profile.html">自己紹介</A><BR>
<BR>
<BR>
<A href="diary.html">日記</A><BR>
</P>
</BODY>
</HTML>
```



《第３図》第2図のファイルを見てみると，こんな感じに。「03.html」は「top.html」と同じ階層に位置しているのでフォルダ名を書く必要はありません

ものです。異なるサーバーにデータの蓄積されている他サイトとのリンクには，もちろんこの書き方しか使えません。

もう一つの「相対パス」は，自サイト内リンクの場合に，リンク元のファイルとリンク先のファイルの相対的な位置関係だけを踏まえて書くものです。ここで言う位置関係とは「階層」と呼ばれるもので，ファイルを格納してあるフォルダの上下関係を指します。

同じフォルダ内にあるファイル同士のリンクならば「同階層」ですので，タグには<a href="index.html">のように，ファイル名だけを書けばOKです。また，リンク元のファイルから見て「下位」のフォルダにあるファイルとリンクする場合は，そのフォルダ名も書き，例えば<a href="diary/top.html">のようにします。逆にリンク元のファイルから一つ「上位」のフォルダにあるファイルとリンクする場合は，<a href="../index.html">のように，「../」で一つ上位であることを示します。二つ上位のファイルの場合は<a href="../../index.html">のように，「../」を二つ書きます。（**第2図，第３図，リスト1，リスト2**）

このように，相対パスはフォルダ内の階層をしっかり把握していないと書けないタグです。しかし，めんどくさいからといって自サイト内リンクにも絶対パスを使うのはおすすめできません。というのは，最近のインターネット事情として，プロバイダを変更する可能性を考えた時，相対パスで書いておけばフォルダをコピーするだけですむ「引っ越し」が，絶対パスの場合はサイト内のすべてのリンクを書き換えるハメになるからです。ホームページの運営では，ファイル数が100を超えるのは実はかなりあっという間の出来事です。その100のリンクをすべて変えるなんて恐ろしい事態は，絶対に避けるべきですね。（**第4図**）

さらに，画像を貼りつけるためのタグ<img src="shiraki.gif">も，画像リンクの相対パスがベターということができます。絶対パスで書くと，例えば<img src="http://www.abc.ne.jp/shiraki/img/shiraki.gif">のようになりますが，自サイト内リンクでは相対パスにしましょう。

《リスト2》第3図の画面のソースリスト

```
<HTML>
<HEAD>
<TITLE>白木ミホのホームページ</TITLE>
</HEAD>
<BODY bgcolor="#ffffff" text="#000000" link="#ff0000" vlink="#000099" alink="#0000ff">
<P>
<FONT size="+2">白木ミホの日記</FONT><BR>
<BR>
<BR>
4月1日（月）　ウソぷー<BR>
<BR>
4月1日と聞いて年度始めだとか，仕事がタイヘン，というのはトシヨリくさいので，エイプリルフ
ールだ，と喜んでみる。訳して「4月バカ」。しかし，ウソをつくというのはけっこうアタマの出来
を問われるものだ。私は以前，「結婚しました～」というハガキをウソで出したことがあるのだが，
少なからぬ人に真に受けられ，非常識のレッテルを貼られたことがある。<BR>
<BR>
<BR>
過去の日記はこちら<BR>
<BR>
今年分<BR>
<BR>
<A href="diary/03.html">3月</A>　<A href="diary/02.html">2月</A>　<A
href="diary/01.html">1月</A><BR>
<BR>
2001年分<BR>
<BR>
<A href="diary/2001/12.html">12月</A>　<A href="diary/2001/11.html">11月</A>　<A
href="diary/2001/10.html">10月</A>　<A href="diary/2001/09.html">9月</A>　<A
href="diary/2001/08.html">8月</A>　<A href="diary/2001/07.html">7月</A>　<A
href="diary/2001/06.html">6月</A>　<A href="diary/2001/05.html">5月</A>　<A
href="diary/2001/04.html">4月</A>　<A href="diary/2001/03.html">3月</A>　<A
href="diary/2001/02.html">2月</A>　<A href="diary/2001/01.html">1月</A><BR>
</P>
</BODY>
</HTML>
```



《第4図》いずれやってくる「お引越し」でも，相対リンクなら変更は起きません

自サイト内リンクいろいろ

他ページへリンクする

これは前述のとおり，リンクの基本中の基本です。（**リスト1**）

同ページ内の所定の場所へリンクする

ページのトップでなく，途中を指定してリンクを張ることができます。内容の多いページで上からいちいち見るのは面倒な時に，見たい項目のあるところへ「飛ぶ」リンクです。

タグは，リンク元の <a href ="#　"> タグと，リンク先を表す目印役のタグの二つを使います。目印タグにもがつきますが，文字列は挟む必要はありません。

目印の名前は英数半角文字で命名し，一つのページ内にいくつでも指定できます。また，長いページのいちばん下に，「このページのトップへ戻る」と書かれていることがありますが，あれも機能は同じで，ページの先頭に「"#top"」のような目印をつけたうえで，<a href="#top">などのように指定してジャンプさせています。（**第5図，リスト3**）

他ページの所定の場所へリンクする

他ページへリンクすると同時に，その指定の場所へジャンプするタグです。

「#」の前にファイル名を，「#」の後にジャンプ先の目印の名前を書きこみます。（**第6図，リスト4**）

新ウインドウを開いて表示するリンク

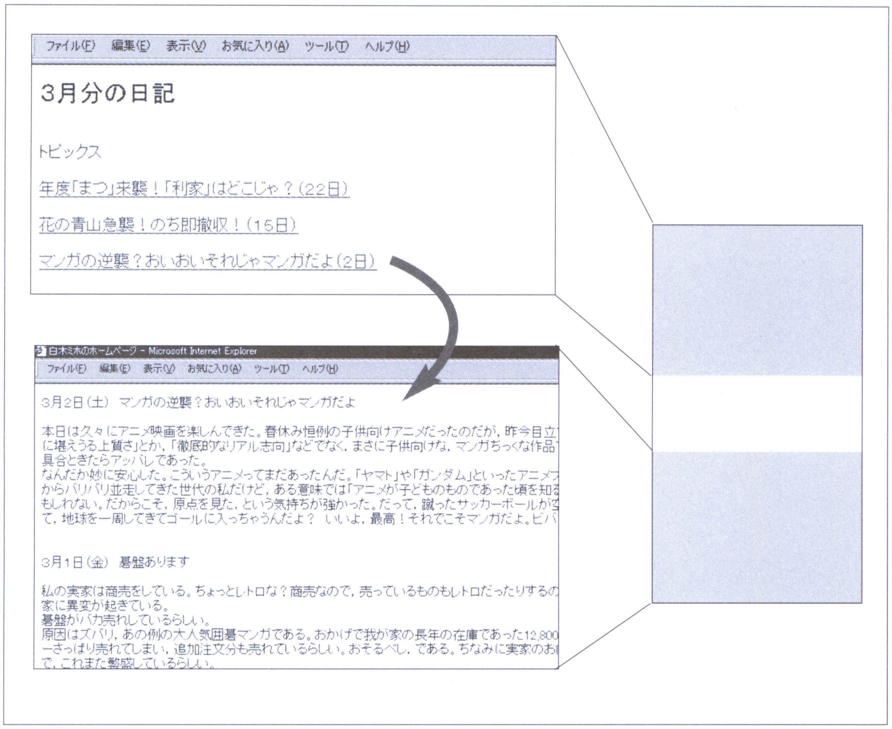
リンク先の表示場所（＝ターゲット）を設定する，「ターゲットオプション」と呼ばれる機能です。「 blank」と書けばリンク元のファイルを表示したウィンドウを残したまま，新しいウィンドウが開いてリンク先が表示されます。（**第7図，リスト5**）

他サイトへのリンクを貼るけれど，自分のホームページの表示も残したい，といった場合や，付属的なデータをちょっと表示したあと，すぐに本題に戻りたい，という場合にこの機能を使うとたいへん便利ですが，あまりたくさん使うといちいち閉じるのが面倒になったり，ツールバーにいつのまにか大量のウィンドウがたまってしまって，パソコンの動きが鈍くなったりする点は不便といえます。

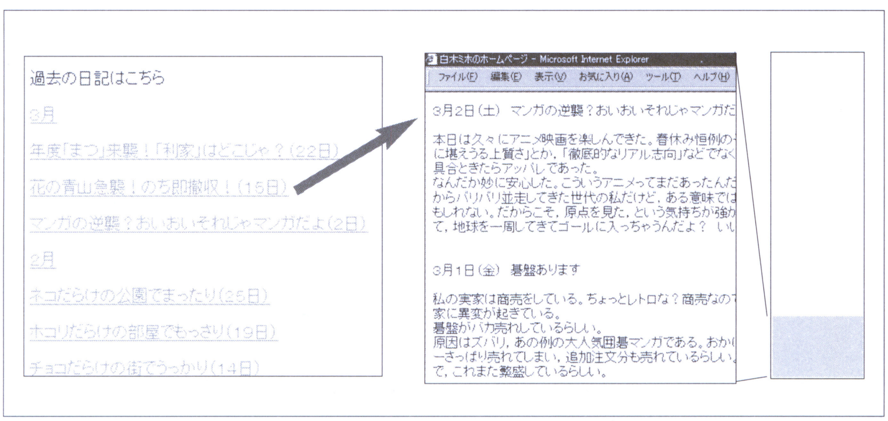
同じウィンドウにリンク先ファイルを表示する指定は，「" self"」と書きます。

メールソフトが立ちあがるリンク

ファイルではなく，メールソフトが呼



《第５図》30日分の日記をアタマから，ただ読ませるのは芸がないので，目次を作ってページ内でジャンプできるようにした例です。どちらの画像も「03.html」ファイルの一部です



《第６図》第5図の方法より一手間省く！　各月の目次を日記のトップページに載せて，指定ページの指定の場所へ一気にジャンプ!

び出されるリンクです。クリック一つでメールソフトが起動し，送信先アドレスも自動的に入力された形で出ます。（**第8図，リスト6**）

逆説？　隠しページ

常識から言えば，ファイルはどこかしらとリンクしなければ誰の目にも触れませんが，これを逆手にとり，リンクでは行けない「秘密のページ」を作る方法もあります。つまり，アドレスを直接入力する以外に閲覧する方法のないページをわざと作る，これを「裏ページ」「隠しページ」などと呼びます。悪意で行なうのは論外ですが，例えばホームページの「オモテ」にクイズを出題し，正解した人にだけ解説「ウラ」ページのアドレスを教える，といった細工は目先が変わって面白かったりします。

ですが，せっかく？隠したファイルが検索機能などによって発見されてしまっては困りますので，そういう時は検索除けの護符タグをつけましょう。例えば，<meta name="robots" content ="noindex,nofollow">を<head>タグの中に書き込めばＯＫ。こうすれば検索にひっかからず，リンクの状況をチェックされることもなく，完全に「隠れ」ることができます。もちろん，ウラページに限らず，オモテのページでも有効なタグです。



《第７図》リンクをクリックすると新しいページが開く「ポップアップ方式」。目次ページも残っているのでたいへん便利ですが，使いすぎるとブラウザだらけになっちゃって重くなります。注意!



《第８図》クリック一つでメールソフトが立ちあがるmailtoリンクは，送信先にアドレスも記入済なのでとっても便利

リンクを張る場所

さて，ここまでリンクタグの「書き方」を解説してきましたが，今度はリンクを張る場所について見て行きましょう。

文字に貼るリンク

<a href>とタグの間に文字を挟むと，その文字をクリックすることでリンクができるようになる，これがリンクの基本ですね。これまでの解説図で作ってきたリンクはどれもこの方法でした。

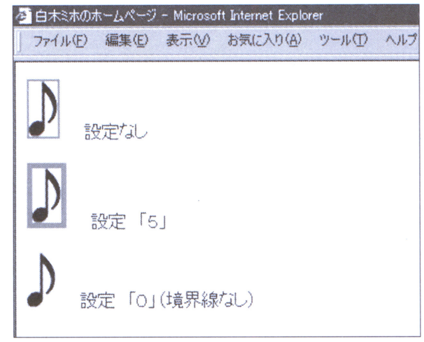
リンクは，実は空白にも，半角英数のピリオド「.」などごく小さいものにも張ることができます。しかし，そんなことをするのはユニバーサルデザイン以前のいじわるとしか思えません。遊び心でやっているホームページをたまに見かけますが，いい感じはしないので，正しい管理人を目指すキミはマネしないでくださいね。

注・文字列リンクの下線を消さない

ユニバーサルデザインの観点からもう

《リスト3》第5図の画面のソースリスト

```
<HTML>
<HEAD>
<TITLE>白木ミホのホームページ</TITLE>
</HEAD>
<BODY bgcolor="#ffffff" text="#000000" link="#ff0000" vlink="#000099" alink="#0000ff">
<P>
<a name="#top"></A>
<FONT size="+2">３月分の日記</FONT><BR>
<BR>
<BR>
トピックス<BR>
<BR>
<A href="#22.htm">年度「まつ」来襲！「利家」はどこじゃ？（２２日）</A><BR>
<BR>
<A href="#15.htm">花の青山急襲！のち即撤収！（１５日）</A><BR>
<BR>
<A href="#02.htm">マンガの逆襲？おいおいそれじゃマンガだよ（２日）</A><BR>
<BR>
<BR>
3月31日（日）　Say, Hyoooo!<BR>
<BR>
出かけ先で突然のカミナリ＆大雨！にみまわれビックリ。何がツライといって，今日おろしたての革ジャケットを着て来ていたことが大ショック。<BR>
と，そこに救いの神登場。車で家まで送ってくれることになり，きゃっほーいてなもんや三度笠である。<BR>
同じく傘なしの人とともに乗せてもらい，雨のバス停を尻目にるんるんと街道を行く。<BR>
「駅に着いたらぜんぜん道ぬれてなかったりしてね～」と笑っていたら，なんとある三叉路を境に突然地面がまっしろ！<BR>
雹が降っていた。。。ドビックリ。<BR>
<BR>
<BR>
3月30日（土）　日本昔話？<BR>
<BR>
なんだか最近いいことないぞジブン，という思いが神に通じたらしく，本日は突然いいことが我が身に降りかかった。<BR>
なんと，道を歩いていて，おむすびをもらった。<BR>
いやはや日本昔話を地で行く展開になってしまったが，正体はリニューアルオープンしたおむすびやさんがやっていた無料キャンペーンだった。本来ならその店で商品として売られているものをひょいとタダでもらってしまうというのはなんだか恐縮である。ちなみにたいへんうまかった。<BR>
おむすびをもらう，というのは昔話においては誠意の象徴とか，幸運の物質化とか，物語の始まりの暗示など，「おむすび」以上の意味がこめられているのが普通だ。いいことあるといいな。</P>
<P>
<a name="#02"></A>
3月2日（土）　マンガの逆襲？おいおいそれじゃマンガだよ<BR>
<BR>
本日は久々にアニメ映画を楽しんできた。春休み恒例の子供向けアニメだったのだが，昨今目立つ「大人の鑑賞に堪えうる上質さ」とか，「徹底的なリアル志向」などでなく，まさに子供向けな，マンガちっくな作品で，そのつきぬけ具合ときたらアッパレであった。<BR>
なんだか妙に安心した。こういうアニメってまだあったんだ。「ヤマト」や「ガンダム」といったアニメブームの出発点からバリバリ並走してきた世代の私だけど，ある意味では「アニメが子どものものであった頃を知る」最後の世代かもしれない。だからこそ，原点を見た，という気持ちが強かった。
だって，蹴ったサッカーボールが空高く突き抜けて，地球を一周してきてゴールに入っちゃうんだよ？　いいよ，最高！それでこそマンガだよ。ビバ！<BR>
<BR>
<BR>
3月1日（金）　碁盤あります<BR>
<BR>
私の実家は商売をしている。ちょっとレトロな？商売なので，売っているものもレトロだったりするのだが，そんな実家に異変が起きている。<BR>
碁盤がバカ売れしているらしい。<BR>
原因はズバリ，あの例の大人気囲碁マンガである。おかげで我が家の長年の在庫であった12,800円の碁盤がきれーさっぱり売れてしまい，追加注文分も売れているらしい。おそるべし，である。ちなみに実家のお向かいは碁会所で，これまた繁盛しているらしい。
</P>
<P>
<A href="#top.htm">▲このページのトップへ戻る</A>
</P>
</BODY>
</HTML>
```



《第9図》画像リンクにつく境界線はどんな形の画像も四角く囲ってしまうのでうっとうしい時もありますが，とっても簡単に消すことができます

一つ注意してほしいことがあります。

文字列にリンクをつけると，普通はその下に線が現れて，リンクのあることが分かる仕組みになっています。ところがこの下線を消してしまうタグがあります。これを使っているホームページの製作者は，おそらくは他と違った雰囲気になってカッコイイくらいの気持からそうしているのかもしれませんが，ただの不親切ですのでやめましょう。ということでタグはあえてご紹介しません。

画像に張るリンク

<a href>とタグの間に文字でなく，画像リンクを挟みこめば，その画像をクリックすることでリンクできるようになります。リンク機能をもたせた小さな画像はアイコンとも呼びます。もちろん，画像自体は小さくても大きくても設定可能です。

リンクを張った画像の周囲には境界線（ボーダー）が表示されますが，この線の太さを変えたり，または消したりすることもできます。（**第9図，リスト7**）

画像リンクとユニバーサルデザイン

ユニバーサルデザインでは，画像には必ずalt機能をつけることを提唱しています。特にリンクを張った画像は，全盲の人の使用する音声リーダで読み上げる際に，alt機能があればその文字列を読むのですが，ない場合にはアドレスを読み上げます。ですから複数のリンクのあるページでは，「リンク・ファイル名profile.html，リンク・ファイル名diary/top.html，リンク・ファイル名http://www…」と延々続いてかなり聞

《リスト4》第6図の画面のソースリスト

```
<HTML>
<HEAD>
<TITLE>白木ミホのホームページ</TITLE>
</HEAD>
<BODY bgcolor="#ffffff" text="#000000" link="#ff0000"
vlink="#000099" alink="#0000ff">
<P>
<FONT size="+2">白木ミホの日記</FONT><FONT size="+2"><BR>
</FONT><BR>
<BR>
4月1日（月）　ウソぷー<BR>
<BR>
４月１日と聞いて年度始めだとか，仕事がタイヘン，というのはトシヨリくさいので，エイプリルフールだ，と喜んでみる。訳して「４月バカ」。しかし，ウソをつくというのはけっこうアタマの出来を問われるものだ。私は以前，「結婚しました～」というハガキをウソで出したことがあるのだが，少なからぬ人に真に受けられ，非常識のレッテルを貼られたことがある。
<BR>
<BR>
<BR>
過去の日記はこちら<BR>
<BR>
<A href="03.html">３月</A><BR>
<BR>
<A href="03.html#22.htm">年度「まつ」来襲！「利家」はどこじゃ？（２２日）</A><BR>
<BR>
<A href="03.html#15.htm">花の青山急襲！のち即撤収！（１５日）
</A><BR>
<BR>
<A href="03.html#02.htm">マンガの逆襲？おいおいそれじゃマンガだよ
（2日）</A><BR>
<BR>
<A href="02.html">２月</A><BR>
<BR>
<A href="02.html#25.htm">ネコだらけの公園でまったり（２５日）
</A><BR>
<BR>
<A href="02.html#19.htm">ホコリだらけの部屋でもっさり（１９日）
</A><BR>
<BR>
<A href="02.html#14.htm">チョコだらけの街でうっかり（１４日）
</A><BR>
<BR>
<A href="01.html">１月</A>
<BR>
<BR>
<A href="01.html#24.htm">長時間睡眠派？短時間睡眠派？（２４日）
</A><BR>
<BR>
<A href="01.html#18.htm">紅茶派？コーヒー派？（１８日）</A><BR>
<BR>
<A href="01.html#03.htm">年頭のごあいさつ。今年の目標は遅刻をなくそう？（3日）</A><BR>
</P>
</BODY>
</HTML>
```

《リスト5》第7図の画面のソースリスト

```
<HTML>
<HEAD>
<TITLE>白木ミホのホームページ</TITLE>
</HEAD>
<BODY bgcolor="#ffffff" text="#000000" link="#ff0000"
vlink="#000099" alink="#0000ff">
<P>
<FONT size="+2">白木ミホのホームページ</FONT><BR>
<BR>
<BR>
<IMG src="shiraki.gif" alt="百万ドルの笑顔" hspace="10"
align="left">
<A href="self.html" target="_blank">自己紹介</A><BR>
<BR>
<BR>
<A href="diary/top.htm" target="_blank">日記</A><BR>
</P>
</BODY>
</HTML>
```

《リスト6》第8図の画面のソースリスト

```
<HTML>
<HEAD>
<TITLE>白木ミホのホームページ</TITLE>
</HEAD>
<BODY bgcolor="#ffffff" text="#000000" link="#ff0000"
vlink="#000099" alink="#0000ff">
<P>
<FONT size="+2">白木ミホのホームページ</FONT><BR>
<BR>
<BR>
<IMG src="shiraki.gif" alt="百万ドルの笑顔" hspace="10"
 align="left">
<A href="profile.html" target="_blank">自己紹介</A><BR>
<BR>
<BR>
<A href="diary/top.htm" target="_blank">日記</A><BR>
<BR>
<BR>
<A href="mailto:shiraki@abc.ne.jp">メールはこちら</A><BR>
</P>
</BODY>
</HTML>
```

《リスト7》第9図の画面のソースリスト

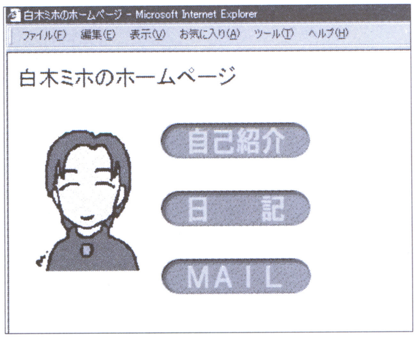
```
<HTML>
<HEAD>
<TITLE>白木ミホのホームページ</TITLE>
</HEAD>
<BODY bgcolor="#ffffff" text="#000000" link="#ff0000"
vlink="#000099" alink="#0000ff">
<P>
<A href="abc.htm">
<IMG src="note.gif"></A>　　設定なし<BR>
<BR>
<A href="def.htm">
<IMG src="note.gif" border="5"></A>　　設定　「5」<BR>
<BR>
<A href="ghi.htm">
<IMG src="note.gif" border="0"></A>　　設定　「0」（境界線なし）
</P>
</BODY>
</HTML>
```

きづらいものになってしまうのです。

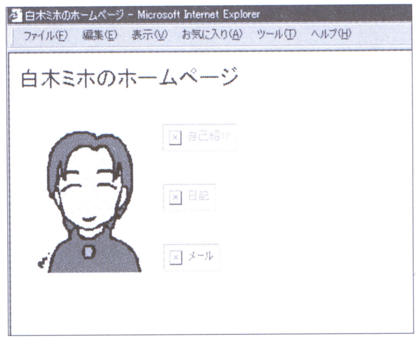
これにきちんとalt機能をつけておけば，「自己紹介，日記…」とすっきりできます。（**第10図，リスト8**）

このalt機能には，画像がリンク切れを起こした，つまりは管理人がチョンボをしたときに，alt文が表示されて，意図したことはとりあえず伝わるという利点もあります。（**第11図**）

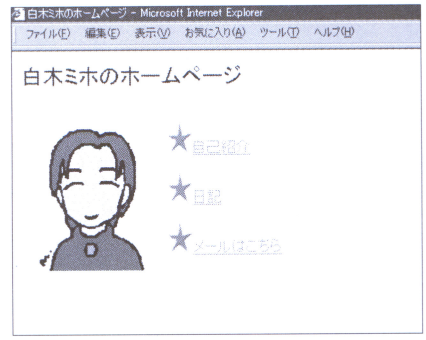
反対に，あえてalt機能に何も書きこまず，空白一つだけを入れるというテクニックもあります。例えば**第12図**はアイコンと文字列を併用し，その両方にリンクを張った例ですが，このアイコンにアイコン自体の説明文やリンク先の解説がついていると，リーダの読み上げ文は「青い星・自己紹介，自己紹介…」と読み，かえって混乱します。ですからここでは**リスト9**のように空白を入れて，「…自己紹介」とテキストだけを読ませるのです。



《第10図》音声リーダはソースが頼りなので，画像の中の文字は読めません。alt機能はユニバーサルデザインの重要ポイントです



《第11図》意外と便利なalt機能。万一リンク切れがあっても笑ってごまかせます??



《第12図》特に重要でないアイコンなどを音声リーダにいちいち読ませない，alt機能の技アリ編。altひとつでキミのサイトは格段に変われるのです

《リスト8》第10図の画面のソースリスト

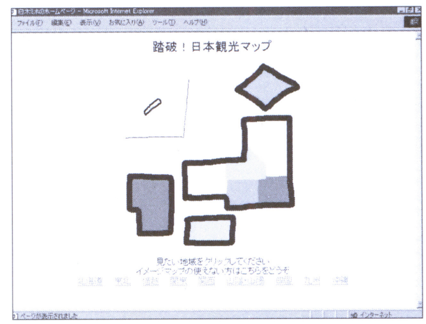
```
<HTML>
<HEAD>
<TITLE>白木ミホのホームページ</TITLE>
</HEAD>
<BODY bgcolor="#ffffff" text="#000000" link="#ff0000" vlink="#000099" alink="#0000ff">
<P>
<FONT size="+2">白木ミホのホームページ</FONT><BR>
<BR>
<BR>
<IMG src="shiraki.gif" alt="百万ドルの笑顔" hspace="10" align="left"><A href="profile.html"
target="_blank"><IMG src="profile.gif" border="0" alt="自己紹介"></A><BR>
<BR>
<BR>
<A href="diary/top.htm" target="_blank"><IMG src="diary.gif" border="0" alt="日記"></A><BR>
<BR>
<BR>
<A href="mailto:shiraki@abc.ne.jp"><IMG src="mail.gif" border="0" alt="メール"></A><BR>
</P>
</BODY>
</HTML>
```

《リスト9》第12図の画面のソースリスト

```
<HTML>
<HEAD>
<TITLE>白木ミホのホームページ</TITLE>
</HEAD>
<BODY bgcolor="#ffffff" text="#000000" link="#ff0000" vlink="#000099" alink="#0000ff">
<P>
<FONT size="+2">白木ミホのホームページ</FONT><BR>
<BR>
<BR>
<IMG src="shiraki.gif" alt="百万ドルの笑顔" hspace="10" align="left"><A href="profile.html"
target="_blank"><IMG src="star.gif" alt=" " border="0">自己紹介</A>
<BR>
<BR>
<A href="diary/top.htm" target="_blank"><IMG src="star.gif" alt=" " border="0">日記</A>
<BR>
<BR>
<A href="mailto:shiraki@abc.ne.jp"><IMG src="star.gif" alt=" " border="0">メールはこちら
</A>
<BR>
</P>
</BODY>
</HTML>
```

画像の一部に張るリンク

イメージマップ，クリッカブルマップ，マッピング機能などと呼ばれるもので，**第13図**のように，画像の一部にリンク機能をもたせてクリックできるようにする機能もあります。ホームページ作成支援ソフトでは比較的簡単にできる機能ですし，たいへん便利なのですが，座標を指定するのは自力ではほぼ無理ですし（**リスト10**），この機能に対応していないブラウザがあることや，そもそも全盲の人には使えないという難点はあります。



《第13図》やっぱり便利なイメージマップ機能。でも画像を使えない人のための配慮は忘れずに!

どうしても使いたいときは，必ず「代替テキスト」と呼ばれる文字列リンクを分かりやすいところにつけるのがユニバーサルデザインの証です。

「戻る」リンクの重要性

「行く」リンクがあればもちろん「戻る」リンクもあります。むしろ使い勝手の良いホームページにはこの「戻る」リンクの充実が欠かせません。

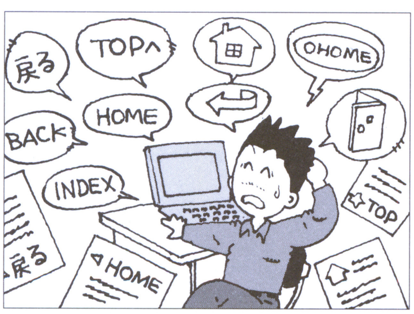
トップページ以外のところに「お気に入り」登録をしている人や，検索でたまたまサイト内のどこかのページを見つけて飛んで来た，という人にもトップページへの道のりを示し，迷子を出さないようにしましょう。

統一性をもたせよう

ひとくちに「戻る」リンクといっても「このページのトップへ戻る」のか，「日記のトップページへ戻る」のか，あるいはその「サイトのトップページへ戻る」のか，それらの機能がごちゃまぜになっているホームページはかなり見かけます。用語や画像，それらの配置の不統一も困

《リスト10》なんじゃこりゃーっ!?という感じのソースを一部ご披露。画像内の座標を示す数字はマウスで囲ったもので，自力での入力はムリ。ホームページビルダー2001の製作例です

```
<HTML>
<HEAD>
<TITLE>白木ミホのホームページ</TITLE>
</HEAD>
<BODY bgcolor="#ffffff" text="#000000" link="#ff0000" vlink="#000099" alink="#0000ff">
<P align="center"><FONT size="+2">踏破！日本観光マップ</FONT><BR>
<BR>
<IMG src="jpnmap.gif" alt=" " hspace="10" usemap="#jpnmap" border="0"><MAP name="jpnmap">
  <AREA shape="poly" coords="267,8,217,49,270,85,328,44,268,5" target="_blank"
href="hokkaidou.htm">
  <AREA shape="poly" coords="230,108,231,153,272,191,274,214,318,212,316,106,230,106"
target="_blank" href="touhoku.htm">
  <AREA shape="poly" coords="229,158,226,182,193,188,196,218,268,217,268,196,228,160"
target="_blank" href="shinetsu.htm">
  <AREA shape="poly" coords="255,222,258,270,318,269,315,220,256,221" target="_blank"
href="kantou.htm">
  <AREA shape="poly" coords="195,225,195,273,251,268,250,223,194,224" target="_blank"
href="kansai.htm">
  <AREA shape="poly" coords="114,194,115,252,188,256,191,188,114,192" target="_blank"
href="san.htm">
  <AREA shape="poly" coords="119,298,118,340,199,342,205,294,121,298" target="_blank"
href="shikoku.htm">
  <AREA shape="poly" coords="9,215,9,275,25,277,29,320,88,319,89,219,8,216" target="_blank"
href="kyusyu.htm">
  <AREA shape="poly" coords="63,67,34,91,40,97,67,73,65,67" target="_blank"
href="okinawa.htm">
  <AREA shape="default" nohref>
</MAP><BR><BR>
見たい地域をクリックしてください<BR>
イメージマップの使えない方はこちらをどうぞ<BR>
<A href="hokkaido.htm" target="_blank">北海道</A>     <A href="touhoku.htm" target="_blank">
東北</A>     <A href="shinetsu.htm" target="_blank">信越</A>     <A href="kanto.htm"
target="_blank">関東</A>     <A href="kansai.htm" target="_blank">関西</A>     <A
href="san.htm" target="_blank">山陰・山陽</A>     <A href="shikoku.htm" target="_blank">四国
</A>     <A href="kyusyu.htm" target="_blank">九州</A>     <A href="okinawa.htm"
target="_blank">沖縄</A></P>
</BODY>
</HTML>
```



《第14図》「戻る」のはタイヘン。ひとつ前のページに戻るつもりがトップページに飛ばされて…なんてことは避けましょう

りものです。サイト全体を統一する基準を必ず設けておきましょう。(第14図)

ツリー構造を考える（サイトデザイン）

ホームページは，最初にとりあえず「立ち上げ」てしまって，それから日々少しずつページを増やしていくのがいい，またそれが出来るメディアだという話を4月号でもしましたが，ここまでを読んだキミは，やっぱりある程度の計画性は要るんだなあと思っているかもしれませんね。実は「戻る」リンクなどの整合性や，ファイルを格納するフォルダの構造を考えることだけは，計画性を要求されるところなのです。

構造は，枝分かれする形から「ツリー(木)」と呼ばれています。また，そのツリー構造を決めることを「サイトデザイン」と呼んでいます。デザインは人さまざまで，実に個性的な形を発明している人もいます。長い時間をかけて関わり，育てていくサイトですから，基本を理解した後は自分の分かりやすい形を追求していってください。

例1　フォルダを全く作らない

小規模なサイトでは，いっそフォルダなしというのも手です。プロバイダから割り当てられた個人フォルダの中にただ放りこむだけ，簡単明瞭です。相対パスで悩む心配もゼロです。

ただ，サイトが大きくなってくると，膨大なファイルにてこずりますし，ファイルのネーミングにきちんとしたポリシーがないとぐちゃぐちゃになる可能性が高いです。ただ漫然とつけるのでなく，たとえばコーナーのトップページにあたるファイル名はすべて「dia」「pro」など頭3文字で統一するとか，それを受けて日記関連のファイルはすべて「dia」で始めて，「dia0204」(日記ファイル・2002年・4月)「dia0112」(同・2001年・12月）と年月を数字で表すなど，統一規格を作っておけば，一覧表を見ても関連ファイルがまとまって表示され，また内容もわかりやすくなるでしょう。

例2　画像フォルダしか作らない

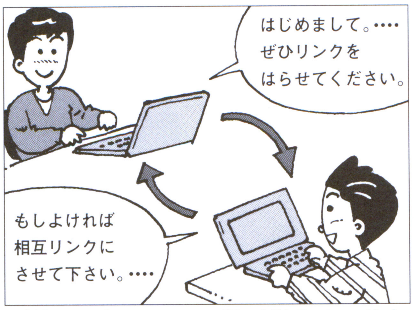
htmlファイルと画像ファイルをきっちり住み分ける方法です。これもまたサイトの規模が大きくなると面倒ですが，上記のような統一規格をもったネーミングなどを心がけておけば，中規模くらいまでなら充分対応できます。もちろん階層も単純で迷いません。

例3　各コーナーごとにフォルダを作る

サイトの中にいろいろなコーナー作る予定のある人や，探し物のヘタな人・キライな人にはこの方法が無難だと思います。ファイルの数が増えてもフォルダごとにきちんと分けてあれば探すのが簡単ですし，新コーナーを増やすのもラクです。反対にコーナーを消滅させる時にも，フォルダ一つを削除すればすみますので，該当ファイルを一つ一つ探し出して削除する手間を省けます。

ただ，フォルダを作ると階層が出現するので，あまり深層化させると相対パスを書く時に困ります。特にあちこちのページで背景画像を共通して使う時は，作成のたびに画像フォルダとの位置関係を数えなくてはなりません。

この問題をクリアするために，各コーナーの中に画像フォルダを設け，画像は一つのコーナーの中でだけ共有する，あるいは共通で使用する画像は極力小さな容量のものにして，各コーナー内の画像フォルダの中にコピーして入れてしまう，という方法を採っている人もいます。



《第15図》アクセス数が増える，共存共栄のリンクはマナーを守って!

デザインのコツは

どの形をとるにせよ，ファイル名・フォルダ名のつけ方が，今後のサイト運営の運命を決定づけます。基本としては分かりやすい名前をつける，これに限ります。もちろん「自分だけが分かればいい」のですから，そこはアイディア次第です。頭3文字の法則を発展させて，ファイル名の文字数を見れば階層の位置が分かる，なんてスグレモノのシステムを開発する人もいますし，古いコーナーから順に01，02，と番号を振って，すべてのファイル名を番号で管理してしまう人もいます。

フォルダは，階層が深くなりすぎないように注意する，また，フォルダを新設する際のルール作りも重要です。読んだ本の感想文をＳＦ，時代物などのジャンルで区切るのか，はたまたマンガ，小説などの形式で区切るのかは，あとあとの分類に影響します。

他のホームページへリンクを張る

さて，ようやくリンクの集大成，他サイト間リンクにこぎつけました。他サイト間リンクはのように絶対パスで書きます。当然ですね。

インターネットの世界は，このリンクをたどって流れていくことから，ネットサーフィンという単語を生み出しました。しかし，目に見えるのが自分のパソコンのモニタだけであっても，その向こうには他のサイトを作っている生身の製作者さんがいるわけですから，当然守らなねばならないマナーも存在します。

マナー1　メールでごあいさつ

《リスト11》私のサイトのここを見て!　ブラウザ画面上には見えない文字で，検索ロボットに対しこちらからアピール。「見つけてもらえる」可能性がぐっと広がります。ちなみに画面は第3図のindex.htmlと一緒になります

```
<HTML>
<HEAD>
<TITLE>白木ミホのホームページ</TITLE>
<meta name="keywords" content="diary, profile, 日記, 自己紹介">
</HEAD>
<BODY bgcolor="#ffffff" text="#000000" link="#ff0000" vlink="#000099" alink="#0000ff">
<P>
<FONT size="+2">白木ミホのホームページ</FONT><BR>
<BR>
<BR>
<IMG src="shiraki.gif" alt="百万ドルの笑顔" hspace="10" align="left"><A
href="profile.html">自己紹介</A><BR>
<BR>
<BR>
<A href="diary.html">日記</A><BR>
</P>
</BODY>
</HTML>
```

たいていのホームページでは，リンクに際してのルールを決めてありますので，まずはそれを守ることが基本です。

注意したい点は，「リンクフリー」というのは，誰でもリンクしていいですよということですが，勝手にしていいというわけではない，ということです。たとえそのホームページにリンクフリーと書いてあっても，リンクを張る場合はまずサイトの管理人さん宛にメールなどでお知らせしましょう。「メールなどでの告知も不要です」と書いてない限り，「ごあいさつとお願い」はネット社会の必需品と思ってください。メールもただ「リンク張らせて」だけではあまりに唐突ですので，簡単に自己紹介や，なぜリンクを張りたいのかなどを書くべきでしょう。もちろん，自分のホームページのアドレス明記も忘れずに。中には「リンクを張らせていただきました。事後報告ですみません」なんてすっとんだメールをよこす人もいますが，ていねいに見えて実は思いきりマナー違反です。謝る前にお伺いを立てましょう。

マナー2　カウンタに愛を

最近のホームページは，閲覧者数を表示する「カウンタ」をつけているところがほとんどです。また，カウンタはだいたいトップページに設けられています。もしもキミがトップページ以外のところへリンクを張ってしまったら，リンク先の管理人さんには訪問者の増加が把握できなくなってしまいます。これは寂しいことです。カウンタがくるくる回ってさえいれば実社会でイヤなことがあっても気を取り直せる，というのがサイト管理人の赤裸々な真実？なのですから，どうか愛をもって接してやってください。

ということで，リンクはカウンタのあるトップページに張り，「○○のコーナー内に注目情報があります」といった解説を文章化してリンクに添えるのがスマートです。

マナー3　相互リンク

ある日，知らない人からメールが来て，どきどきしながら見てみたらリンク依頼だった……ホームページ管理人にとっては実に嬉しいことです。自分の作ったホームページを見てくれて，気に入ってくれた人がいたんだ，という幸せです。

そんな時は「相互リンク」をぜひ提案してください。(第15図)

相互という名のとおり，相手から自分にリンクを張ってもらう，そのお返しに今度は自分のところから相手のところへリンクを張り，お互いにリンクしあうということです。相互リンクを張ることで，お互いのページについてくれている「常連さん」たちを引き込みあい盛りあがれれば，ホームページがますます活気づくでしょう。

検索サイトに登録

ホームページは，出るところには出て，目立つべきところでは目立たなければ，存在そのものを知ってもらうチャンスはたいへん少ないものです。

「検索サイト」はそんなホームページのための「電話帳」です。ジャンルやキーワードで検索をかければ該当するホームページを発見することができます。今やインターネットをしない人にもその名を広めた「Yahoo!」はこの検索サイトの代表ですが，他にも「inforseek」や「goo」などホームページ全般を扱う大規模サイトから，あるジャンルに限った小規模検索サイト，果ては検索サイトを集めた検索サイトが存在するほどにその数は増えています。

検索サイトには，情報を申請・登録するものと，ロボット検索といって機械で自動的に検索をかけるものの２種類があり，現在は登録型のサイトでもロボットを併用している場合がほとんどです。

そこで，ロボット検索により効率よく「引っかけて」もらうために，キーワードをあらかじめページ内に書きこんでおくのもいい方法です。それも，ロボットが本文よりも優先的に読み取る<head>タグ内の<meta>タグに書くのがミソです。（リスト11）

このタグを逆の意味＝検索されないために使ったのが，前述のロボット検索除けの護符です。

まとめ

ホームページとはリンクである，と言い切ってもいいくらいに，リンクとは重要な存在です。htmlファイル１枚きりのホームページは，おそらくないでしょうし，多くのアクセス数を稼ぎ出している人気ページが検索サイトの一つにも登録していないということは，おそらくないはずです。半面，やれ誰それのホームページが引っ越したと言っては書き換え，やれ友達の友達のサイトとお義理で相互リンクして，といった面倒も実はあったりするのですが，華やかなサイトというものが多彩なお友達に恵まれている点は，現実社会とまったく同じです。

本連載も折り返し点に来て，ホームページの基本をおさえたという感じです。次回は発展編のお待ちかね「画像」をテーマにお送りします。ではでは。

コラム　検索サイト・リング・同盟に登録しよう　白木ミホ

まだ見ぬホームページを探す，あるいは欲しい情報を検索する，そんなホームページ探し専門のホームページが，検索サイト，あるいはリンク集，リングと呼ばれるものです。同時に情報の発信者＝ホームページの管理人にとっては，より多くの人にサイトを見てもらうための強力なアイテムとも言えるでしょう。

●検索サイト

キーワードやジャンルを手がかりにホームページを検索するホームページで，なんといっても最大手は，あの『Yahoo！Japan』。サイト管理人が自己申請した内容をヤフー側でチェックし，登録する方式と，特定の単語を含むホームページのアドレスを自動的に拾い上げる検索ロボット・プログラムによる自動表示の２種類が併用されています。

『All About Japan』は，「その道のプロ」による解説や，イチオシサイト紹介のある検索サイトで，案内人の顔が見える親しみやすさがあります。

●リンク集

特定のジャンルに特化したこだわりで勝負している検索サイトで，『Ahooサーチエンジン』は，「アホなサイト」と「笑えるサイト」しか紹介していない強烈どころです。また，『縦書きサイト普及委員会』は，文字どおり縦書きページをもつサイトだけを登録し，日本語の魅力をスマートに主張するリンク集です。

●リング

特定の趣旨にそったホームページをナビゲーションバーで案内するプログラムを使ったリンク集で，登録サイトが一周可能なひとつなぎの輪になるため，この名前がついています。案内バーには，「次」「前」「ランダム」などのボタンがあり，クリックで別のサイトに飛べます。このプログラムを無償配布しているウェブリングJapanの公式ページには，登録リングの検索サイトもあります。

10代のサイト製作者によるホームページを集めた『TeensSiteRING-New-』は，これからどんどん加盟者を増やしてほしい，赤丸要チェックリング。面白いところでは，もぐらたたきゲームがあることを条件にした『組長のもぐらたたきリング』が注目。ひとくちにもぐらたたきといっても，阪神ファンサイトの「ジャビットたたき」など，遊び心は実に多彩です。

●同盟

こちらは，一定の趣旨に賛同した人たちが名乗りをあげ，掲示板やチャットなどで交流することを目的としています。ホームページではなく個人（もちろんハンドル名と呼ばれる仮名）を登録するので，ホームページをもっていない人でも参加できるのが大きな特長。その同盟専門の検索サイト『同盟ナビ』にはすでに5000以上の登録があるそうですが，製作者自らが立ち上げた「血液型Ａ型同盟」や「両利き同盟」など，思わず入りたくなるユニークさが競われています。

サイトアドレス	
Yahoo! Japan	http://www.yahoo.co.jp/
All About Japan	http://allabout.co.jp/
Ahooサーチエンジン	http://www.jp-paradise.net/ahoo
縦書きサイト普及委員会	http://isweb28.infoseek.co.jp/computer/bobodori/index.html
ウェブリングJapan	http://www.webring.ne.jp/
TeensSiteRING-New-	http://usagi.tadaima.com/mikito/teens/index.html
組長のもぐらたたきリング	http://www.kurumi.sakura.ne.jp/~kumicyou/
同盟ナビ	http://www2.tok2.com/home/auau111/

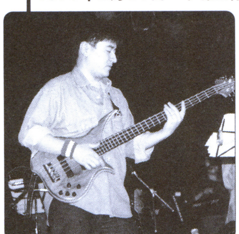
BM基礎講座

ベーシックプログラミング教室

第3回「サブルーチンとキー入力」

講師紹介

川上朝幸（Kawakami Tomoyuki）
1971年8月31日　大阪府生まれ



音楽系専門学校卒業後 93年カプコン入社。サウンドエンジニアとして，大ヒット作"ストリートファイター２"，"ヴァンパイア"，"ロックマン"シリーズをはじめ"バイオハザード"，"ディノクライシス"など，多くのサウンド制作を経て00年退職，翌01年"スタジオ スターノーツ"を起業。サウンドエンジニアとしての活動を軸にプログラミング，パソコンスクール講師，ホームページ，ビデオ制作，アマチュアバンド「フロントーサ」のアルバム制作（夏発売予定），自身のバンド「スターノーツ」のライブなど，幅広い活動を行っている。

STARNOTES
http://www.starnotes.jp
basic@starnotes.jp

この講座では，Windows上で実行することができ，誰でも容易に入手することができるフリーソフトを利用します。

この講座はBASICとして基本的なN88 BASICに基づいた内容ですので，あらゆるベーシックで応用することができます。

☆フリーで登録されているベーシックプログラミング・ソフトをいくつか紹介します。

●N88互換BASIC for Windows95
<http://www.vector.co.jp/soft/win95/prog/se055956.html>

●ActiveBasic
<http://www.vector.co.jp/soft/win95/prog/se117283.html>

講座内ではこれらを互換BASICと呼ぶことにします。

みなさん，こんにちは。ベーシックプログラミング教室も早いもので，もう3回目です。今回はちょっとしたゲームをサンプルプログラムとして用意してみました。ゲームそのものは極めて単純なものですが，シンプルなものほど，ハマリやすいものです。そして，そのゲームのプログラムを見ていくと結構奥が深いものだったりします。さあ，今回も楽しくプログラミングしましょう！！

プログラムを入力しよう！

それでは早速，互換BASICを起動して次のプログラムを入力してみましょう。ちょっと長いプログラムですが，入力ミスをしないように頑張りましょう！

プログラムを入力したら早速実行してみましょう。
実行画面に，

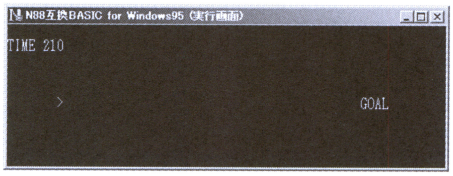
HYPER DASH!!
START! >> HIT [SPACE] KEY

＊サンプルプログラム10

```
100 *TOP
110 CLS
120 X=5:Y=5:TSUGI$="1KEY":TC=0:MTC=1000:G=50
130 PRINT "HYPER DASH!!"
140 PRINT "START! >> HIT [SPACE] KEY"
150 WHILE A$<>" ":A$=INKEY$:WEND
160 CLS:LOCATE G,Y:PRINT "GOAL":GOSUB *HYOJI
170 WHILE X<G AND TC<MTC
180   A$=INKEY$:TC=TC+1
190   IF A$="1" AND TSUGI$="1KEY" THEN GOSUB *HYOJI:TSUGI$="2KEY"
200   IF A$="2" AND TSUGI$="2KEY" THEN GOSUB *HYOJI:TSUGI$="1KEY"
210   LOCATE 0,1:PRINT "TIME";TC
220 WEND
230 IF HI<MTC-TC THEN HI=MTC-TC
240 PRINT "HI-SCORE";HI;"  SCORE";MTC-TC
250 PRINT "REPLAY?(Y or N)"
260 *KEYIN
270 A$ = INKEY$
280 IF A$="Y" THEN *TOP
290 IF A$="N" THEN END
300 GOTO *KEYIN
310 *HYOJI
320 X=X+1
330 LOCATE X,Y:PRINT ">"
340 RETURN
```

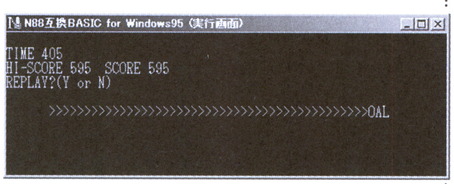
と表示されるので［SPACE］キーを押してみましょう。

画面左上に，TIMEと表示されその横で数字がカウントしています。その下に＞，その右にGOAL と表示されています。



▲プログラムの実行画面

この状態でキーボード（テンキー）の[1]と[2]を交互に押すと，画面中央の＞が右に動きだします。[1][2]を素速く押すことで移動速度も上がります。＞が右のGOALの文字に達するか，TIMEが1000になったら終了です。スコアが表示され，もう一度やるか終了するか選択することができます。



▲プログラムの実行画面

GOALに，どれだけ速く到達することができるかを競うゲームですね。速く到達すればするほど得点が高くなります。終了するまでのハイスコアも記憶されます。シンプルですが，友人とコナミのハイパーオリンピックというゲームで必死になって遊んでいたときのことを思い出してしまう懐かしいゲームです。

プログラム解説

はじめに，このプログラムはマシンの性能によって実行速度が変わります。高速なマシンになるほどプログラムの処理速度が速くなるので，TIMEのカウントも速くなってしまいます。TIMEが1000になるまでに，どうしてもGOALにたどり着けない場合は，120行目のG=50 の値を小さくしてみてください。キャラクター">"が移動する距離が短くなるので，時間内にGOALにたどり着けるようになります。Celeron 600MHzのマシンではちょうどよい感じなのですが，Pentium4 1.8GHzのマシンで実行するとかなり厳しかったので調整が必要でした。

ひとしきり遊んだところで，プログラムを見てみましょう。初めて見る語がたくさんありますね。

100＊TOP

これはラベルと言い，行に名前をつけるものです。行番号の次に＊(アスタリスク)を入力してから変数名と同じように名前を付けます。これまでIF～THENやGOTOといった命令で行番号を指定してジャンプしていましたが，このラベルを利用してもジャンプすることができます。280行目の，IF A＄="Y" THEN＊TOPは，IF条件式を満たせば＊TOP にジャンプするプログラムになっています。ラベルの便利なところは，行番号が変わっても関連する命令を変更しなくてもよいことです。

(例)

```
10 'ラベルがない場合
20 PRINT "こんにちは"
30 GOTO 10
```

↓行番号を100からはじめたプログラムに変更した場合。

```
100 'ラベルがない場合
200 PRINT "こんにちは"
300 GOTO 100
```

と，最後のGOTO 10 をGOTO 100に変える必要があります。

```
10 ＊TOP  'ラベルがある場合
20 PRINT "こんにちは"
30 GOTO＊TOP
```

↓行番号を100からはじめたプログラムに変更した場合。

```
100 ＊TOP  'ラベルがある場合
200 PRINT "こんにちは"
300 GOTO＊TOP
```

この場合，行番号を変更しても他に変更する箇所はありません。ラベルはとても便利なので今後どんどん使って行きましょう！

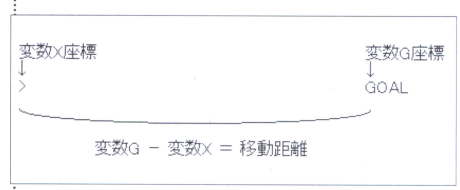
120 X=5:Y=5:TSUGI$="1KEY":TC=0:MTC=1000:G=50

ここでは，プログラム中に使用する変数の値をあらかじめセットしています。こうしておくことで，プログラムの改造や調整をすることが楽になります。(本来は変数ごとに行を分けて，REMでコメントをつけておくと，より見やすくなるのですが紙面の都合でまとめています。)

ここでセットしている変数

X, Y	キャラクター“>”を表示する座標。
TSUGI$	キー入力の判別用。
TC	タイマーの値。得点(スコア)計算にも使います。
MTC	タイマーの最大値。得点(スコア)計算にも使います。
G	キャラクターの到達するゴールの位置(横座標)。

X,Yの値を変更すると，キャラクター">"のスタート座標が変わります。Xの初期値を変数Gよりも大きくしてしまうとエラーになるので注意してください。変数Gと変数Xの値の差が，キャラクターの移動距離になります。



はじめにGの値を変えることでマシン性能による差を調整する説明をしましたが，Gを変えずにXの値を大きくしても移動距離が短くなるので同じように調整できますね。残りの変数は各行の解説の際に説明します。

150 WHILE A$<>" ":A$=INKEY$:WEND

ここでは，二つ新しい語が出てきています。わかりやすくするために，この行を分解します。

```
150 WHILE A$<>" "
152   A$=INKEY$
154 WEND
```

まずWHILE～WEND命令です。これは先月号に出てきた，FOR～NEXTとIF～THENを組み合わせたような命

令で，条件による繰り返し処理を行います。

```
WHILE 条件式
  繰り返す処理
WEND
```

条件式が成り立つ間，WHILE〜WENDの間にある処理を繰り返すというものです。条件式には，IF 条件式 THEN で使う条件式と同じものを使うことができます。ここでは，152行目のA$=INKEY$という処理をA$<>" " つまりA$が“ ”（半角スペース）でない間，繰り返すということになります。

ここで新しくINKEY$という関数が出てきます。関数という言葉も初めてですね。簡単に説明すると，ある情報を元に何らかの処理を行い，その結果を返すものを関数といいます。このINKEY$は，キーボードから入力された文字情報を読み取って一つの文字列を返す関数です。A$=INKEY$は，キーボードからの入力情報を変数A$に代入します。もしキーボードが押されていなければ，""という空の文字列が代入されます。

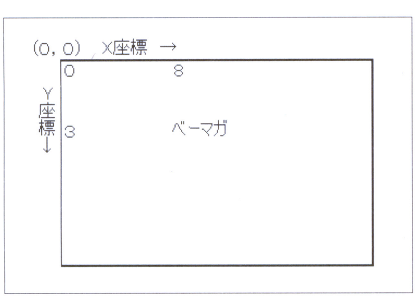
つまり150〜154行目は，スペースキーが押されるまでの間キーボードからの入力情報をチェックしなさいという処理になります。スペースキーを押すと繰り返しを終了し，次の処理（160行目）を実行します。

```
160 CLS:LOCATE G,Y:PRINT "GOAL":GOSUB *HYOJI
```

LOCATE命令は先月号で紹介しましたが，カーソルの位置を指定する命令です。

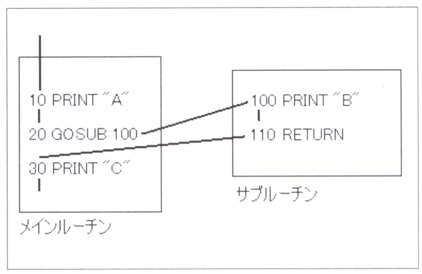
例）

```
10 LOCATE 8,3
20 PRINT"ベーマガ"
```



(座標は半角単位で指定します。X,Y座標の文字数の制限については，来月号で詳しく説明します。)

ここでは，LOCATE G,Y と，X座標に変数Gを使用し“GOAL”と表示しています。次に新しくGOSUBという命令がでてきます。GOSUBは340行目のRETURNという命令とともに使われます。GOSUBは指定したラベルや行番号にジャンプし，RETURNが出てくるとGOSUB命令の次の処理を行います。次の図を見てください。



このプログラムの場合，行番号10,20,100,110,30 の順で実行されます。このGOSUBを使ったプログラムのジャンプ先からRETURNまでのプログラムをサブルーチンといいます。プログラム中でよく使われる処理をまとめたもので，効率の良いプログラムを作っていく上でとても便利なものです。

160行目のGOSUB *HYOJIでは，310から340行目までのサブルーチンを実行するという命令になります。310〜340行目のサブルーチンについては次で説明します。

```
170 WHILE X<G AND TC<MTC
180    A$=INKEY$:TC=TC+1
190    IF A$="1" AND TSUGI$="1KEY" THEN GOSUB *HYOJI:TSUGI$="2KEY"
200    IF A$="2" AND TSUGI$="2KEY" THEN GOSUB *HYOJI:TSUGI$="1KEY"
210    LOCATE 0,1:PRINT "TIME";TC
220 WEND
```

これは先ほど説明したWHILE〜WENDを用いたループ（繰り返し）処理を行っています。170行目の条件式X<G AND TC<MTC を満たす間処理を続けます。つまり，キャラクターがゴールに到達していない状態（X<G）でかつ（AND）制限時間をオーバーしていない状態（TC<MTC）であればループするということになります。

180行目のINKEY$でキーボードからの入力をチェックします。TC=TC+1でタイマー値に1を加算していき，210行目のPRINT命令でTCの値を表示しています。TCの値は，ループの度に1ずつ加算されるので，この数字の増え方を見ることでWHILE〜WEND間のループ速度がわかりますね。この速度はマシンの性能によって大きく差が出ます。

190，200行目は，このプログラムの最も重要な部分で，[1][2][1][2]と交互にキーボードを押した場合にキャラクターを動かすという処理を行っています。変数TSUGI$が次に押すキーの設定をしています。TSUGI$の初期値を

【ワンポイント】〜繰り返し処理

次のプログラムも150行目のプログラムと同じ処理を行います。

```
150 *KEYIN
152 A$=INKEY$
154 IF A$<>" " THEN *KEYIN
```

スペースが押されなければ，*KEYINというラベルにジャンプする処理を行うことで，繰り返しの処理を行います。

プログラムの260行目〜300行目のようにGOTO文を使って繰り返しを行うこともできますね。260行目〜300行目を150行目のように，WHILE~WEND命令を使って書くこともできます。

```
260 WHILE A$<>"N"
270    A$=INKEY$
280    IF A$="Y" THEN *TOP
290 WEND
300 END
```

WHILE~WENDの間の処理を，A$が"N"でない間繰り返すプログラムになります。"N"が入力されると繰り返しを終了し，300行目のENDを実行します。

このように繰り返したい状況に応じていろいろな方法でプログラムできるようになりましょう。150行目の場合は，WHILE~WENDを使うことで，とてもシンプルにまとまっていますね。

120行目でTSUGI$="1KEY"としているので，まず[1]キーの入力待ちになります。[1]キーが入力されるとTSUGI$="2KEY"となり[2]キーの入力待ちになります。[2]キーが入力されるとTSUGI$="1KEY"となり再び[1]キーの入力待ちになります。

○TSUGI$="1KEY"のとき
[1]キーを押すと190行目のIF条件式が成立し，GOSUB ＊HYOJIを実行しTSUGI$="2KEY"とします。

○TSUGI$="2KEY"のとき
[2]キーを押すと200行目のIF条件式が成立し，GOSUB ＊HYOJIを実行しTSUGI$="1KEY"とします。

190行目をA$="7"，200行目$="8"にすると，[7]キー[8]キーを交互に入力するプログラムに変わります。

次にGOSUB ＊HYOJI のサブルーチンです。＊HYOJIというラベルにジャンプしRETURNで戻ってきます。

```
310 ＊HYOJI
320 X=X+1
330 LOCATE X,Y:PRINT ">"
340 RETURN
```

190，200行目でキー入力があった場合に実行されるサブルーチンですが，X=X+1でキャラクター">"の表示位置を右に半角1文字分ずらして表示します。

```
X座標→123456789
       >
       >>←X=X+1
```

これを繰り返すことで右に動いているように見えますね。

190，200行目で同じ処理を行っているので，サブルーチンにしてGOSUB ＊HYOJIで共通に呼び出す（コール）ようになっています。160行目からもコールされていますね。

最後に，230,240行目，

```
230 IF HI<MTC-TC THEN HI=MTC-TC
240 PRINT "HI-SCORE";HI;" SCORE";MTC-TC
```

WHILE～WENDのループを抜けだしたら実行されます。ゴールに到着するかタイムオーバーになった場合ですね。230行目は，ハイスコアの処理を行っています。MTC-TCは，タイマーの最大値（MTC）からWHILE～WENDを抜けだしたときのタイマー値（TC）を引いた値を求めています。速くゴールすればよりTCの値が小さくなるので，MTC-TCの値が大きくなります。つまり点数が高くなります。点数は240行目のSCOREで表示されます。HIはハイスコアを保存する変数です。230行目で，もしHI<MTC-TC の条件式が成立すればHIを更新し240行目で表示します。

WHILE～WENDを使った条件によるループ処理，GOSUB～RETURNを使ったサブルーチン処理，INKEY$関数を使ったキーボードからの入力，とゲーム制作にはどれも欠かすことのできないものばかり！！ちょっと難しく感じる部分もあるかもしれませんが，プログラムを作ってみたり，改造したりして，これらの命令に触れていくうちに，少しずつ理解できるようになってきます。

慌てず焦らずがんばりましょう！！

サンプルプログラムを改造しよう！

今回のサンプルプログラムの改造例を紹介するのでチャレンジしてみましょう！！この改造例の中にも，いくつか新しい命令語が出てきますので，しっかりチェックしておいてください。

○改造その1～キャラクターの移動

サンプルプログラムでは，">"が右に移動するときに痕跡が残ってしまいますね。より移動している感じを出したいので，この痕跡を消しましょう！330行目のPRINT ">" の">"の前に半角のスペースを入力しましょう！この半角スペースがあることでX=X+1で表示を右にずらしたときに古い位置にあった">"が消去されます。

```
X座標→123456789
       " >"
         ↑スペースで消去
        " >"←X=X+1
```

```
330 LOCATE X,Y:PRINT " >"
```

このままだとゴールしたときにGOALのGOまで消えてしまいます。半角スペースを追加したため半角1文字分はみ出てしまったのです。170行目でX<G-1とすることで，これまで同様Gの文字が消えたときにゴールする設定に修正することができます。

```
170 WHILE X<G-1 AND TC<MTC
```

○改造その2～カラフルなキャラクター

移動する度に色が変わるカラフルなキャラクターにしてみましょう！COLOR命令を使います。COLORの後に色ナンバーを入れることで，PRINT命令などで表示する文字の色を変えることができます。色ナンバーは，0=黒，1=青，2=赤，3=紫，4=緑，5=水色，6=黄，7=白です。COLOR 2 と設定すると，その後に実行するPRINT命令などの文字は全て赤で表示されます。

```
325 COLOR 2
```

キャラクターの色が赤に変わったと思います。一度設定したCOLOR設定は次に設定し直すまで有効なので，キャラクターだけでなくTIMEなどの表示にも影響されてしまい，あまりきれいではないですね。キャラクター以外の表示は元のまま白にしましょう。

```
330 LOCATE X,Y:PRINT " >":COLOR 7
```

330行目の後ろにCOLOR 7 と白色の設定を入れましょう。これでキャラクター以外の文字は白のままですね。初期設定は白ということで，110行目にも入れておくとよいでしょう。

```
110 CLS:COLOR 7
```

さて，ここでは色をカラフルに変え

たいので，320行目を次のように変えてみましょう！

```
325 COLOR (X MOD 7)+1
```

X MOD 7は，4月号に出てきた計算式ですね。Xを7で割ったときの余りを求める計算です。7で割ったときの余りは0～6になります。しかしこのままでは0=黒が表示されてしまい，見えなくなることがあるので，+1をすることで1～7の色を使用するようにしています。

○改造その3～エンディング時のキー処理

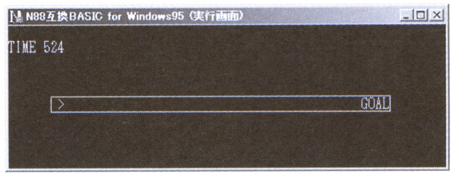
260～300行目では，Yを入力するとプログラムをはじめ（*TOP）から実行し，Nを入力すると終了するという分岐処理を行っていますが，ひとつ欠点があります。キーボードから入力すY,Nは，半角大文字でなければいけないということです。キーボードの設定（CAPS LOCKの状態）はプレイする人によって異なるので，半角小文字のy，nの入力も判断する設定にしておくほうがよいですね。

```
280 IF A$="Y" OR A$="y" THEN *TOP
290 IF A$="N" OR A$="n" THEN END
```

○改造その4～線を引いてみましょう。

```
165 LINE (X*8,Y*16-1)-((G+4)*8,(Y+1)*16),7,B
```

これを入力して実行するとキャラクター">"の移動する部分を白線で囲むことができます。100m競争のトラックに見えますね（苦笑）。



▲プログラムの実行画面

LINEという命令がでてきました。これは画面上に線を引く命令です。

```
LINE (X1,Y1)-(X2,Y2),C,[B or BF]
```

（X1,Y1）で線を引き始める始点をセットし（X2,Y2）で終点の座標をセットします。Cで表示する色を設定します。この色ナンバーはCOLOR命令と同じになります。

```
LINE (10,10)-(100,100),5
```

とすると，水色で斜めの線を引くことができましたね。Cの後に，Bをつけると，始点と終点を対角線とした四角を書くことができます。Bの変わりに，BFをつけると水色で塗りつぶした四角を書くことができます。

```
LINE (10,10)-(100,100),5,B
LINE (10,10)-(100,100),5,BF
```

ポイントとなるのが，ここで使用しているX1,Y1,X2,Y2の座標です。これはLOCATEで使う半角単位の座標とは違いドット単位の座標になります。ドットとは画面に表示することのできる最小の点のことをいいます。皆さんがウインドウズで使っている画面サイズはいくつですか？800×600であれば横に800個の点と縦に600個の点を表示できる画面になります。1024×768になると，さらに細かくなりますね。LINE命令では，このドットの座標を指定します。

追加した165行目の各座標の値は，一見複雑な計算を行っているように見えますが，これはLOCATEで使うX,Y座標をドット座標に変換する計算を行っているからです。半角文字は横8ドット，縦16ドットで構成されています。Y*16-1と-1ドット少なくしているのは，キャラクターとくっついてしまわないように調整しているからです。

ここまでの改造をしたプログラム（サンプルプログラム11）はHPに掲載していますので，うまくいかなかった人はダウンロードしてください。

次回予告

今回は盛りだくさんの内容になったので練習問題のスペースがなくなってしまいましたね。今月はお休みします。来月は4回目，いよいよ折返し地点ですね！次は画面に絵を描いてみよう！ということでグラフィック関連の命令語を中心にお届けします！

この講座で使用しているプログラムデータは下記のホームページでダウンロードすることができます。
http://www.beeplus.or.jp/bm/basic/
☆BM基礎講座「ベーシックプログラミング教室」
制作 スタジオスターノーツ
http://www.starnotes.jp/
協力 有限会社 ビープラス
http://www.beeplus.or.jp/

ゲームを作ってみよう!

舞台は,星音(ほしね)高校パソコン研究部。4人の部員たちが初めてのゲーム制作にチャレンジ！彼らもベーシックプログラミング教室を読みゲームを作ろう!!と意欲漫々!!第3回となる今回,だんだん難しくなっていくプログラムに,メインプログラマーの「ちぃさん」はちょっと大変のようです。

登場人物

「ちぃさん」(ちぃ)　・・・2年生。部長。パソコンは詳しいけど，プログラミングははじめて。
「みぃちゃん」(みぃ)・・・2年生。唯一の女子。その大胆な発想にみんないつも驚き。
「よっさん」(よっ)　・・・3年生。頼りになる先輩。とてもやさしいアニキです。自称ゲーマー。
「ゆうちゃん」(ゆう)・・・1年生。パソコンにはあまり詳しくないけど，音楽ならまかせて。

～星音高校，パソコン研究部部室～

ちぃ：「ふー！できたー！先月の宿題だった，“プレイヤーの年齢でパラメータを変化させる！”ができたよ！よっさん先輩に見てもらわないと。」

ゆう：「今日は『マイコンBASICマガジン』の発売日ですよ～早速買ってきましたよ！はい。」

ちぃ：「お！ありがとう。よし，えっと～ベーシックプログラミング教室と…あったあった。」

ゆう：「今回はどうですか？」

ちぃ：「お！こんなことができるんだぁ～ふむふむ。それからそれから…」

ゆう：「すっかりはまってますね～。そういえば，みぃちゃんとよっさん先輩が見当たらないなぁ。」

ちぃ：「おー！これは使える！それからそれから…」

みぃ：「やっほー！みんな元気～！」

よっ：「まいど。」

ゆう：「あ，よっさん先輩とみぃちゃん。こんにちはー！どこに行ってたんですか？」

みぃ：「マップを作るために校舎の中をいろいろと見て周ってたの。」

よっ：「今回のゲームの舞台はこの星音高校やからな。」

ゆう：「あ！ゲームの内容が決まったんですね。もしかして仮で作ってた『星音高校殺人事件』ですか？」

みぃ：「タイトルはまだ考えてないけど，内容的には『星音高校の謎』って感じかな。」

ゆう：「『星音高校の謎』かぁ～。どんなストーリーですか？」

みぃ：「それは，もう少しお楽しみに！」

よっ：「ちぃさん。調子はどないでっか？」

ちぃ：「ふむふむ…。え，あ，お，よ，よっさん先輩じゃないですか～！びっくりした～！」

よっ：「おいおい，そんなにびっくりせんでもええやん(笑)」

ちぃ：「よっさん先輩，先月号の宿題できましたよ！見てください…こうやって…」

よっ：「ちぃさん，あの宿題もうええよ。」

ちぃ：「え？」

みぃ：「仕様をちょっと変更して年齢というパラメータは入れないことにしたのよ。ごめんね。」

ちぃ：「そんなぁ…」

よっ：「まあゲーム作っていく上では，企画や仕様変更はよくあることやから，そう落ち込まない。ゲーム会社でも発売寸前まで出来上がっていたゲームを突然ボツにすることだってあるんやから。」

ゆう：「え，そうなんですかぁ。」

ちぃ：「まあ，それに比べればまだマシか…」

よっ：「あと性別のパラメーターも入れないことにしたから。」

ちぃ：「りょ，りょうかいしました…。トホホ…」

みぃ：「ちぃさん，ゲームのメイン画面をちょっと変更したので修正してくれる？」

よっ：「今月号のベーシック教室で習った命令を使ったら，先月のプログラムをよりすっきりできるで。」

ちぃ：「あ，はい。わかりました。やってみます。あれ？…よっさん先輩いつの間にベーマガ読んだんだろう？まあいいか。」

～1時間後～

ゆう：「ちぃさん黙々とプログラム書いていますね。もう1時間になりますよ。」

みぃ：「今は声かけないほうがいいわよ。噛みつかれるわよ。」

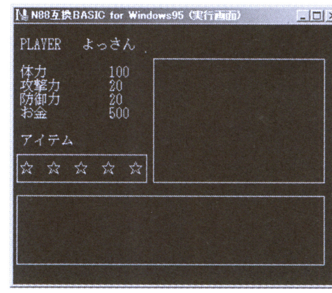
ゆう：「コ・コワイ…」

ちぃ：「よし！できたー！これでどうかな？ん？噛みつく？」

よっ：「早速実行してみるで。名前を入れてと…」

ゆう：「あ，画面が出た！先月より線がきれいになってる！」

みぃ：「いい感じじゃない。」

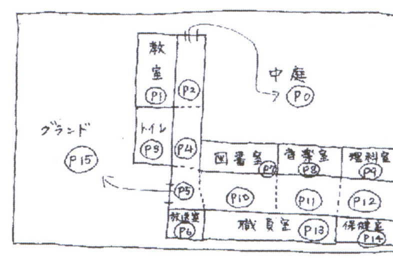


ちぃ：「今回習った命令をいっぱい使ってみました。最後のLINE命令のところはちょっとややこしかったけど，これで1020～1035行目の変数の値を変更するだけでLINEで引く線のサイズも変えることができるので後々ずいぶん楽になると思います。」

よっ：「うん。なかなかいい感じやな。ちぃさんも，だいぶプログラム慣れてきた見たいやな。」

ちぃ：「あ，ありがとうございます！もっともっとがんばります！」

みぃ：「はい。じゃ次これお願いね。今回のゲームのマップよ。」



ちぃ：「OK。えっと…実際の星音高校とはちょっと違うんだね。」

みぃ：「ゲーム用にちょっとシンプルにしてみたの。」

ちぃ：「P1とかP2っていうのは？」

みぃ：「プレイヤーが移動できる場所なんだけど，プログラムしやすいようにIDナンバーをつけてみたの。」

ちぃ：「ほーなるほどね。これも，みぃちゃんが考えたの？」

みぃ：「これは，よっさん先輩のアドバイス。」

ちぃ：「さすが，よっさん先輩！」

よっ：「このマップの中を自由に動けるようにプログラムできるかな？」

ちぃ：「え！？えっと…と，とりあえずやってみます。」

よっ：「絵はまだできてないから文字を表示して，移動してるのがわかればいいよ。」

ちぃ：「は，はい。やってみます。」

```
10 '星音高校　パソコン研究部
20 '　　プログラム制作開始　2002/3/1
30 '　　　　　　Version 0.06　　2002/5/1
50 '初期設定
55 PLAYER$ = ""　　'名前
60 HP=100　　　　　'体力
65 ST=20　　　　　 '攻撃力
70 DF=20　　　　　 '防御力
75 GD=500　　　　　'お金
100 CLS
110 PRINT "星音高校の謎（仮）"
120 PRINT
130 WHILE PLAYER$ = ""
135　 INPUT "主人公の名前を入力してください";PLAYER$
140 WEND
150 GOSUB ＊GAMEN
999 END
1000 ＊GAMEN　　　　　　　　　　　　　　　'基本画面
1010 CLS
1020 X1=1:Y1=1　　　　　　　　　　　　　　'基本画面の左上X,Y座標
1025 X2=0:Y2=8:GYO2=2:RETU2=19　　　　　'アイテムウインドウ調整
1030 X3=20:Y3=1:GYO3=9:RETU3=25　　　　'メインウインドウ調整
1035 X4=0:Y4=11:GYO4=5:RETU4=45　　　　'メッセージウインドウ調整
1040 LOCATE X1,Y1:PRINT "PLAYER　";PLAYER$
1050 LOCATE X1,Y1+2:PRINT "体力　　　　　";HP
1060 LOCATE X1,Y1+3:PRINT "攻撃力　　　　";ST
1070 LOCATE X1,Y1+4:PRINT "防御力　　　　";DF
1080 LOCATE X1,Y1+5:PRINT "お金　　　　　";GD
1090 'アイテムウインドウ
1100 LOCATE X1,Y1+Y2-1:PRINT "アイテム"
1105 LOCATE X1,Y1+Y2+1:PRINT "☆　☆　☆　☆　☆"
1110 LINE ((X1+X2)＊8-4,(Y1+Y2)＊16+8)-((X1+X2+RETU2)＊8-4,(Y1+Y2+GYO2)＊16+8),7,B
1120 'メインウインドウ
1130 LINE ((X1+X3)＊8-4,(Y1+Y3)＊16+8)-((X1+X3+RETU3)＊8-4,(Y1+Y3+GYO3)＊16+8),7,B
1140 'メッセージウインドウ
1150 LINE ((X1+X4)＊8-4,(Y1+Y4)＊16+8)-((X1+X4+RETU4)＊8-4,(Y1+Y4+GYO4)＊16+8),7,B
1160 RETURN
```

～再び1時間後～

ゆう：「ちぃさんは，またまた黙々とプログラム書いていますね。」

みぃ：「それよりゆうちゃんこれ食べない？新発売のチョコレート。おいしいよー！」

よっ：「ええなぁ。うまそうやなぁ。」

ゆう：「ちぃさん…まあいいか。僕にもください。」

よっ：「おーこれうまいなぁ。」

みぃ：「でしょ，でしょ！」

よっ：「もっと欲しいなぁ。う～でもこれ以上食べると…」

みぃ：「大丈夫！大丈夫！アブ○○○○で減量してるんでしょ。」

よっ：「な，なんでそれを…知ってるねん」

みぃ：「うふふ」

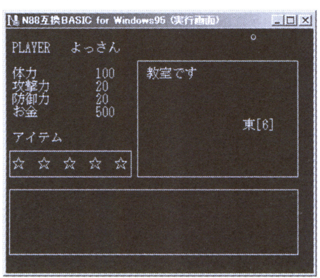
ちぃ：「できた！ん？みんな何やってるの！僕だけほったらかしで…」

ゆう：「いえいえ。ほったらかしなんてそんな…」

みぃ：「そうそう。それはそうとできたプログラム見せてよ。」

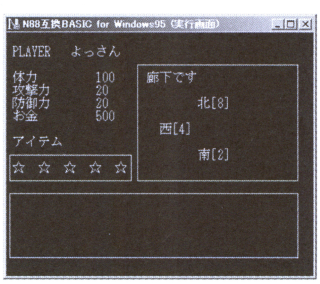
ちぃ：「え，あ，うん。ちょっとプログラムが長くなってきたので全部のマップ動けるようにはなってないんだ。もっといいプログラムの方法があるんだろうけど思いつかなくて。」

よっ：「OK。とりあえず実行してみよう。」



ちぃ：「メインウインドウに，今いる場所と移動できる方向を表示してみました。東[6]となっているので，テンキーの[6]を押すと，教室をでます。」

よっ：「OK！じゃ，[6]と…」



よっ：「お，次は廊下やね。なるほど，ちょっとマップと照らし合わせてみよう。」

ゆう：「教室がP1ですね。教室から東に移動するとP2の廊下。廊下からは北と西と南に移動することができるから…。あ，あってる。[4]を押して西に移動するとまた教室に戻るんですか？」

ちぃ：「そのとおり！よっさん先輩やってみてください。」

よっ：「OK。[4]と…。お，教室に戻った。いいねー。で，今どこまで移動できるんかな？」

ちぃ：「P5の廊下からグランドに出るところまでです。P5で東と南はまだ作ってません。作ってるとだんだんプログラムが長くなってきて

```
10 '星音高校　パソコン研究部
20 '　　プログラム制作開始　2002/3/1
30 '　　　　　Version 0.07　　2002/5/1
50 '初期設定
55 PLAYER$ = ""　　'名前
60 HP=100　　　　'体力
65 ST=20　　　　 '攻撃力
70 DF=20　　　　 '防御力
75 GD=500　　　　'お金
80 XM=1:YM=13　　'メッセージ表示位置
85 XW=22:YW=3　　'メインウインドウ内のメッセージ表示位置
90 K2=0:K4=0:K6=0:K8=0 'マップ移動情報
95 I=1
100 CLS
110 PRINT "星音高校の謎（仮）"
120 PRINT
130 WHILE PLAYER$ = ""
135　INPUT "主人公の名前を入力してください";PLAYER$
140 WEND
150 GOTO ＊P1　'教室からスタート
159 END
160 ＊P0:GOSUB ＊GAMEN:LOCATE XW,YW:PRINT "中庭です"
170 K2=1:K4=0:K6=0:K8=0:GOSUB ＊MAPIDO
180 WHILE I=1:A$=INKEY$
190　IF A$="2" THEN ＊P2
195 WEND
200 ＊P1:GOSUB ＊GAMEN:LOCATE XW,YW:PRINT "教室です"
210 K2=0:K4=0:K6=1:K8=0:GOSUB ＊MAPIDO
220 WHILE I=1:A$=INKEY$
230　IF A$="6" THEN ＊P2
240 WEND
250 ＊P2:GOSUB ＊GAMEN:LOCATE XW,YW:PRINT "廊下です"
260 K2=1:K4=1:K6=0:K8=1:GOSUB ＊MAPIDO
270 WHILE I=1:A$=INKEY$
280　IF A$="2" THEN ＊P4
290　IF A$="4" THEN ＊P1
300　IF A$="8" THEN ＊P0
310 WEND
320 ＊P3:GOSUB ＊GAMEN:LOCATE XW,YW:PRINT "トイレです"
330 K2=0:K4=0:K6=1:K8=0:GOSUB ＊MAPIDO
340 WHILE I=1:A$=INKEY$
350　IF A$="6" THEN ＊P4
360 WEND
370 ＊P4:GOSUB ＊GAMEN:LOCATE XW,YW:PRINT "トイレ前の廊下です"
380 K2=1:K4=1:K6=0:K8=1:GOSUB ＊MAPIDO
390 WHILE I=1:A$=INKEY$
400　IF A$="2" THEN ＊P5
410　IF A$="4" THEN ＊P3
420　IF A$="8" THEN ＊P2
430 WEND
440 ＊P5:GOSUB ＊GAMEN:LOCATE XW,YW:PRINT "正面ロビーです"
450 K2=0:K4=1:K6=0:K8=1:GOSUB ＊MAPIDO
460 WHILE I=1:A$=INKEY$
470　IF A$="8" THEN ＊P4
480　IF A$="4" THEN ＊P15
510 WEND
520 ＊P15:GOSUB ＊GAMEN:LOCATE XW,YW:PRINT "グランドです"
530 K2=0:K4=0:K6=1:K8=0:GOSUB ＊MAPIDO
540 WHILE I=1:A$=INKEY$
550　IF A$="6" THEN ＊P5
560 WEND
999 END
1000 ＊GAMEN　　　　　　　　　　　　'基本画面
1010 CLS
1020 X1=1:Y1=1　　　　　　　　　　　'基本画面の左上X,Y座標
1025 X2=0:Y2=8:GYO2=2:RETU2=19　　　'アイテムウインドウ調整
1030 X3=20:Y3=1:GYO3=9:RETU3=25　　 'メインウインドウ調整
1035 X4=0:Y4=11:GYO4=5:RETU4=45　　 'メッセージウインドウ調整
1040 LOCATE X1,Y1:PRINT "PLAYER　";PLAYER$
1050 LOCATE X1,Y1+2:PRINT "体力　　　";HP
1060 LOCATE X1,Y1+3:PRINT "攻撃力　　";ST
1070 LOCATE X1,Y1+4:PRINT "防御力　　";DF
1080 LOCATE X1,Y1+5:PRINT "お金　　　";GD
1090 'アイテムウインドウ
1100 LOCATE X1,Y1+Y2-1:PRINT "アイテム"
1105 LOCATE X1,Y1+Y2+1:PRINT "☆　☆　☆　☆　☆"
1110 LINE ((X1+X2)＊8-4,(Y1+Y2)＊16+8)-((X1+X2+RETU2)＊8-4,(Y1+Y2+GYO2)＊16+8),7,B
1120 'メインウインドウ
1130 LINE ((X1+X3)＊8-4,(Y1+Y3)＊16+8)-((X1+X3+RETU3)＊8-4,(Y1+Y3+GYO3)＊16+8),7,B
1140 'メッセージウインドウ
1150 LINE ((X1+X4)＊8-4,(Y1+Y4)＊16+8)-((X1+X4+RETU4)＊8-4,(Y1+Y4+GYO4)＊16+8),7,B
1160 RETURN
1200 ＊MAPIDO
1210 IF K2=1 THEN LOCATE XW+8,YW+6:PRINT "南[2]"
1220 IF K4=1 THEN LOCATE XW+2,YW+4:PRINT "西[4]"
1230 IF K6=1 THEN LOCATE XW+15,YW+4:PRINT "東[6]"
1240 IF K8=1 THEN LOCATE XW+8,YW+2:PRINT "北[8]"
1250 RETURN
```

…もっとうまくまとめる方法がないかと考えていたんです。」

よっ：「プログラム見せて。」

ちぃ：「はい。ちょっと長いですよ。」

ちぃ：「各ポイントで移動できる場所のメッセージを書くとプログラムが長くなるので，こんな感じでサブルーチンを作ってやってみました。」

```
210 K2=0:K4=0:K6=1:K8=0:GOSUB ＊MAPIDO

1200 ＊MAPIDO
1210 IF K2=1 THEN LOCATE XW+8,YW+6:PRINT "南[2]"
1220 IF K4=1 THEN LOCATE XW+2,YW+4:PRINT "西[4]"
1230 IF K6=1 THEN LOCATE XW+15,YW+4:PRINT "東[6]"
1240 IF K8=1 THEN LOCATE XW+8,YW+2:PRINT "北[8]"
1250 RETURN
```

ちぃ：「K2,K4,K6,K8っていう変数を用意してみました。これは各ポイントで移動できる方向を設定するための変数です。K2が南，K4が西，K6が東，K8が北になります。210行目はP1の教室なので，マップを見ると東にしかいけない。だからK6=1 でほかは 0 にします。そして，GOSUB で1200行目の＊MAPIDOサブルーチンにジャンプします。ここでは，K2～K8までの値を見て1の場合だけ，メインウインドウに移動できる方向と使用できるキーを表示するようにしています。」

ゆう：「なるほど！僕にもわかります！各ポイントで毎回説明を書いてると確かに長くなりそうだし，この方法だと簡単ですね。」

よっ：「なかなかいい方法やね。まあ，これももうちょっとよい方法はあるけど，考え方としてはいい感じやね。」

ちぃ：「ありがとうございます。もうちょっと良い方法ってなんだろう？調べてみよう。」

よっ：「で，この部分はこの方法を使ってプログラムをまとめることができたけど，各ポイントにあるINKEY$を使うキー入力の部分が，うまくまとめられへんかったんやね。」

ちぃ：「そうなんです。動くだけならもっと，短いプログラムでできるかなぁと思ったんですが…」

よっ：「同じ動作をするプログラムでもプログラムの書き方はいろいろあるからね。もうちょっとガンバっていろいろ考えてみよう。これは来月号までの宿題やな。うまくまとめられないとこれからどんどんプログラムを追加しないとあかんから大変になってくるぞ。」

ちぃ：「はい。わかりました。」

ゆう：「でも，ようやくゲームらしくなってきましたねー。なんかわくわくしてきました。」

みぃ：「まだまだやりたいことたくさんあるんだけどなー。ちぃさん大丈夫？」

ちぃ：「う，うん，もっともっとがんばるよ！」

みぃ：「ストーリーもだいぶできあがってきてるから，次はそれを組み込んでもらおうかな。」

ゆう：「いよいよですね！！」

ちぃ：「OK！がんばって，おもしろいゲームをつくるぞー！」

一同：「おう！！！」

はじめよう！ Microsoft OfficeでBASICプログラミング No.2

協力：（株）富士通南九州システムエンジニアリング

第2回 画面をつくろう

●文/椛島たけし

先月はらはじまった本連載ですが，今回は画面作成の方法を説明します。もちろん基礎から解説しますので，今月からはじめる読者でも大丈夫です。

先月のポイント

まず先月のポイントは，

・BASICでのプログラムをはじめるにはMicrosoft OfficeがあればOK

・Microsoft OfficeのWord，Excel，PowerPointなど同じVBAでプログラミングできる

・プログラムをはじめるのはカンタン

ということでした。手始めとして，Word，Excel，PowerPointそれぞれでコマンドボタンをクリックすると時刻を表示するプログラムを作成しました(**第1図**)。今月はもう一歩すすんでみます。

画面をつくってみよう

前回の例では，Wordの文書やExcelのワークシート等に直接コマンドボタンを貼り付けていました。「ボタンをクリックすると手紙に季節や時候の挨拶が入る」「ボタンをクリックするとグラフを印刷する」といったプログラムの場合，この方法は便利です(**第2図**)。しかし，かならずしも文書やワークシートに結びつかないプログラムもあります。そのような時は，ユーザー・フォームを作成すると便利です。さっそくユーザー・フォームを作成してみましょう(**第3図**)。

ユーザー・フォームができたら，プログラミングの開始です。先月と同じく，コマンドボタンをクリックすると日時を表示するプログラムを作成してみることにします(**第4図**)。

同じプログラムでも，こちらの方がそれっぽいですね。また，ExcelでもPowerPointでも全く同じ方法でユーザー・フォームをもつプログラムを作成できます。ぜひ，ためしてください。

コントロールとは

ツールボックスには、いろいろな部品が表示されています。この部品をコントロールと呼びます。フォームに使いたいコントロールを貼り付けることで，コントロールを使用できるようになります。したがって，実現したい機能に合ったコントロールを選ぶ必要があるわけです。そこで，どのようなコントロールがあって，どのような機能をもっているか事前に知っておく必要があります。今回は，よく使う代表的なコントロールについて紹介します(**第5図**)。

○コマンドボタン・コントロール(CommandButton)

マウス・クリックなどをきっかけに処理を開始・終了する場合に使用します。ボタンには文字や画像を設定することができます。

○テキストボックス・コントロール(TextBox)

データの入力と表示に使用する最も一般的なコントロールです。

○ラベル・コントロール(Label)

文字データを表示する時に使用します。項目(コントロール)の説明や画面の説明などに頻繁に利用されます。データの直接的な入力ができないのが，テキストボックス・コントロールとの大きな違いです。

○イメージ・コントロール(Image)

イメージ(画像)を表示することができます。ビットマップ，GIF，アイコンなど，数種類の形式の画像フォーマットを利用することができます。

○オプションボタン・コントロール(OptionButton)

複数の項目の中で，どの項目が選択されているか判断するときに使用します。種類などを特定するときに便利です。

○フレーム・コントロール(Frame)

オプションボタンなどのコントロールをグループ分けする時に使用します。また，単なるデザインに使うこともできます。

○チェックボックス・コントロール(CheckBox)

一つの項目について選択/非選択を判断する必要があるときに使用します。いわゆる「複数選択可」の項目などに利用するときには，複数のチェックボックス・コントロールを組み合わせます。

イベントドリブンの考え方

ここでちょっとイベントについて説明します。コードウィンドウに表示されている「Command1_Click」は「コマンドボタンがクリックされた」というイベントを表します。つまり，Command1を実際にクリックしたときの動作(処理)をここに書くわけです(第6図)。このように，「マウス・クリック」「キーボード入力」「エラー発生」などイベント(事象)をきっかけにプログラムを作成する方法を「イベント・ドリブン・プログラミング(イベント駆動型)」と呼びます。Office VBAはイベント・ドリブンです。

それに対して，「～をしたら～をする」という手順を，最初から最後まで順に記述する方法を「手続き型プログラミング」と呼びます。

フォームやコントロールには，あらかじめいろいろなイベントが用意されていて，イベントを使い分けることによってさまざまなプログラムを作成できます。主なイベントを紹介します。

○Click

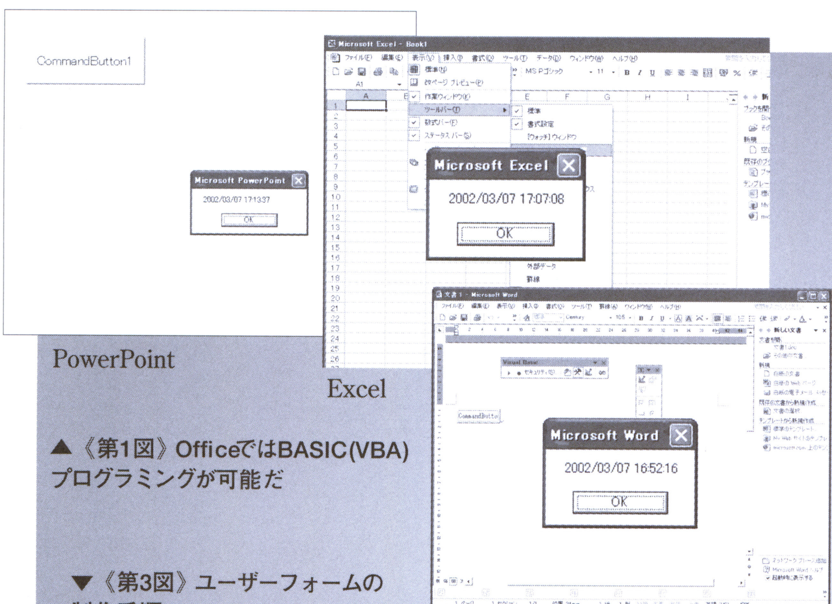
フォームやコントロールがクリックされたとき発生します。最もよく使用するイベントです。

○DblClick

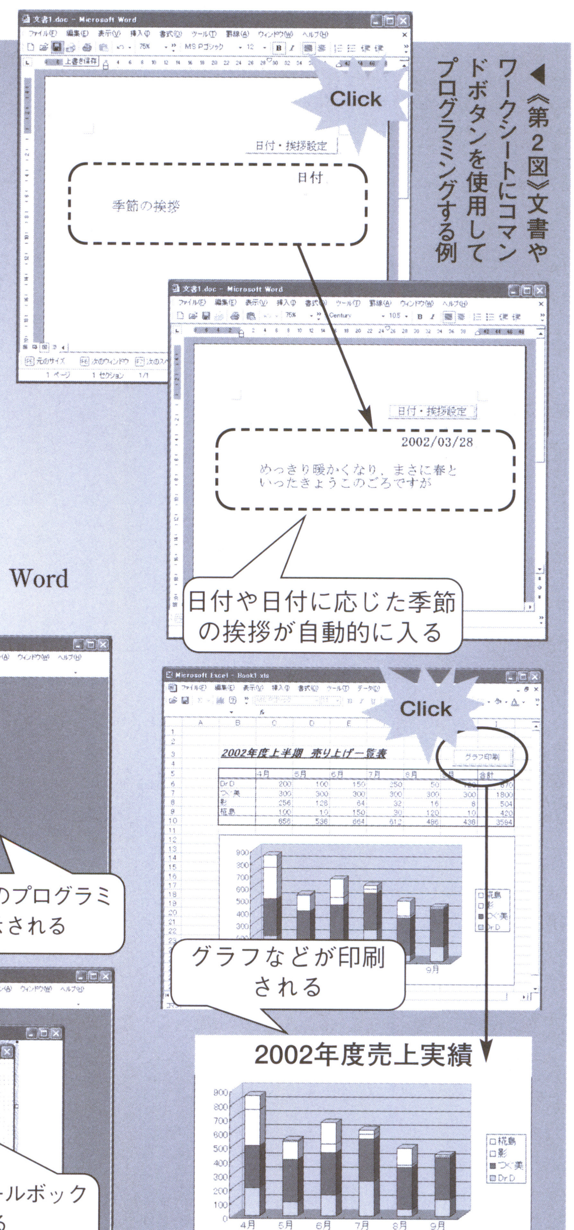
フォームやコントロールがダブルクリックされたとき発生します。プログラムでは意外と使う機会が少ないイベントです。

○Change

コントロールのデータが変更された時に発生します。入力されたデータの確認などに利用します。

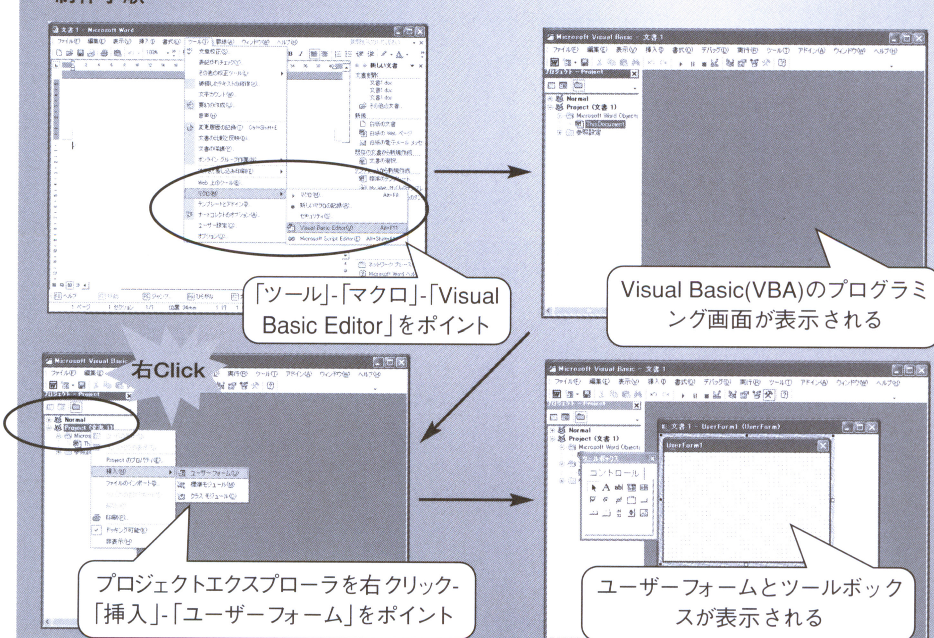


▲《第1図》OfficeではBASIC(VBA)プログラミングが可能だ



◀《第2図》文書やワークシートにコマンドボタンを使用してプログラミングする例

▼《第3図》ユーザーフォームの制作手順



○KeyDown

キーを押した時に発生し，キーコードを受け取ることができます。また，Shiftキー，Ctrlキー，Altキーの状態を取得することも可能です。つまり，「どのキーが押されたか」を知ることができます。

○KeyUp

キーを離した時に発生します。その他の機能はKeyDownと同じです。

○KeyPress

入力可能な文字に対応するキーを押した時に発生し，入力された文字のコードを受け取ることができます。たとえば，アルファベットのみ入力させたい時などに使用できます。ちなみに，矢印キー(方向キー)やTabキーを押しても発生しません。

○Enter

コントロールがフォーカス*を得る直前に発生します。フォーカスを得る動作をキャンセルすることも可能です。

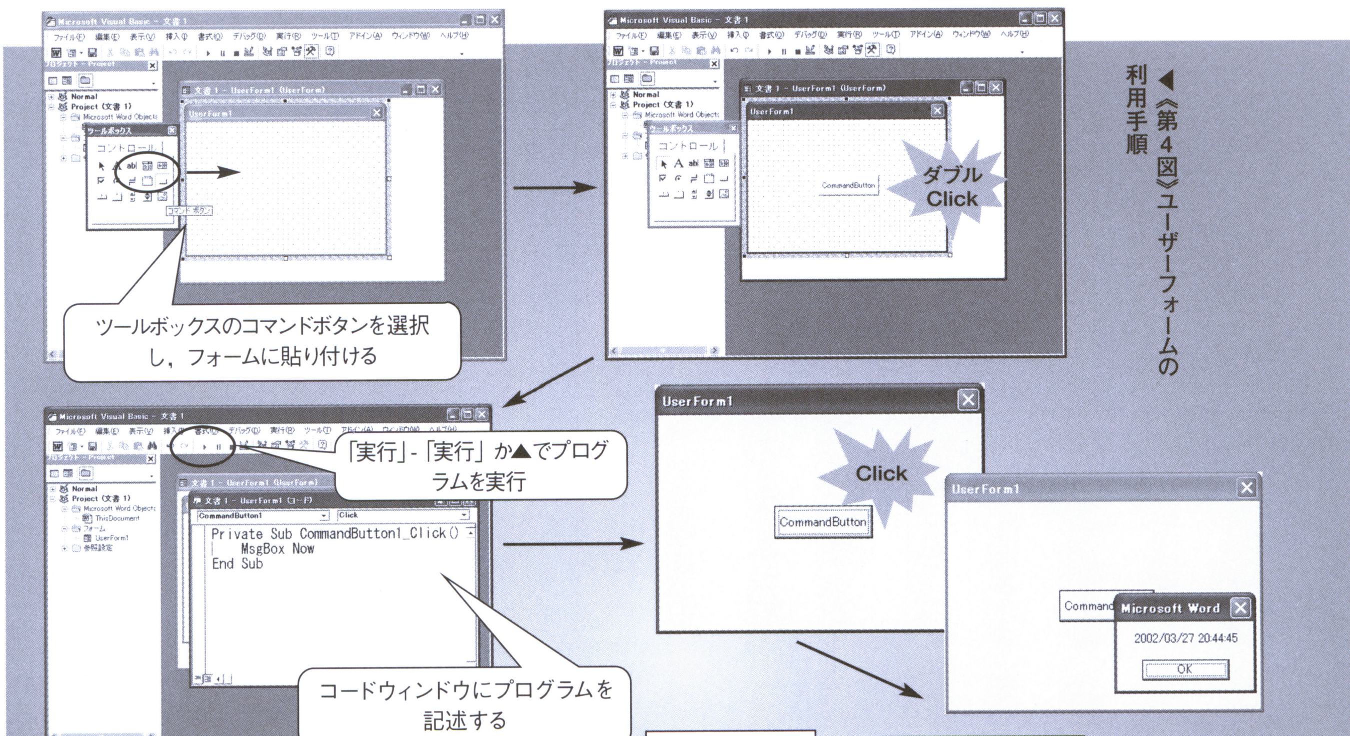
○Exit

コントロールがフォーカスを失う直前に発生します。フォーカスを失う動作をキャンセルすることも可能です。フォーカスを失う際の警告や確認などに利用します。

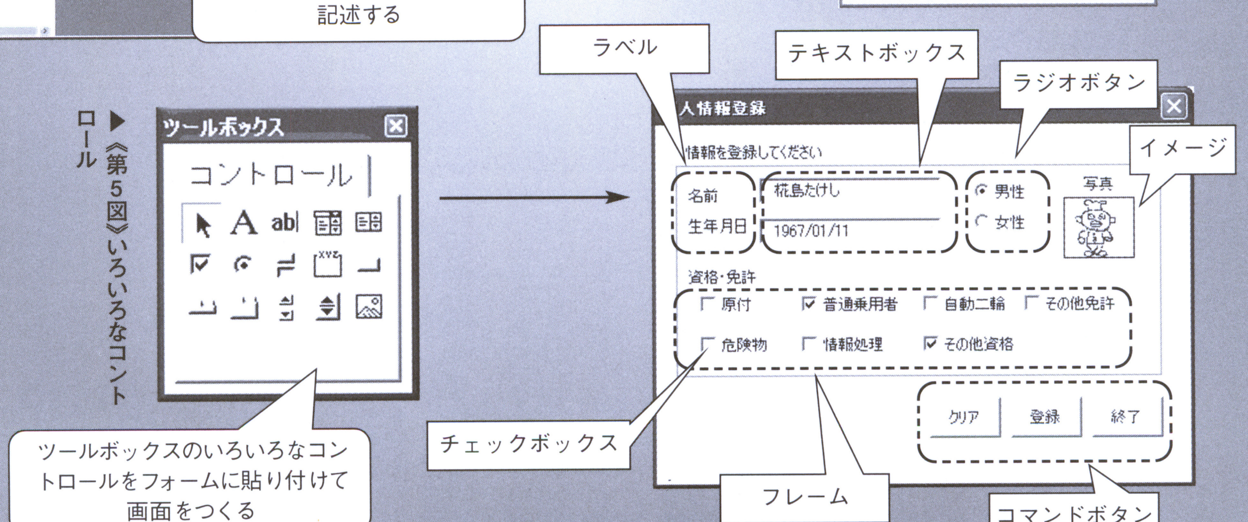
***フォーカス**　マウスやキーボードからの入力を受け取ることができる状態のことです。フォーカスを得ているコントロールは，強調表示になったり，キャプション(説明の文字)が点線で囲まれたりする場合もあります。同時に複数のコントロールがフォーカスをもつことはありません。

さて来月は

2回にわたって，OfficeでBASICプログラミングを行う手順について紹介しました。だいたいのイメージはわかってもらえたでしょうか？。次回からは，BASICの文法も含め，よりプログラムらしい題材に取り組む予定です。



◀《第4図》ユーザーフォームの利用手順



▶《第5図》いろいろなコントロール

ウィンドウの名前

Visual Basicの開発画面には，いろいろなウィンドウがあります(**第7図**)。それぞれの名前と役割について簡単に説明します。

○プロジェクト・エクスプローラ

プロジェクトとして管理されている項目を階層表示しています。フォームやコードを編集したい場合，目的のフォームやコードをプロジェクト・エクスプローラでダブルクリックして編集します。

○ツールボックス

使用可能なコントロールが表示されます。ツールボックスで目的のコントロールを選択し，フォームに貼り付けます。

○コード・ウィンドウ

コード(プログラム)の表示，入力，修正を行います。

○ユーザー・フォーム・ウィンドウ

フォームを編集するウィンドウです。コントロールの配置や大きさの変更をマウスで行うことができます。

○プロパティ・ウィンドウ

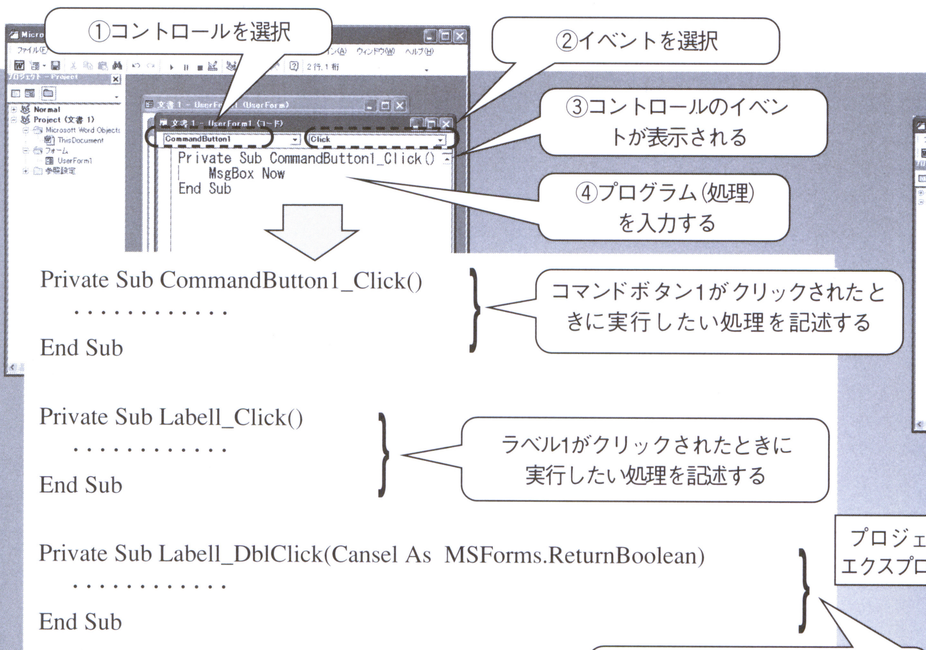
選択されているフォームやコントロールの，プロパティ(属性)を表示します。設定値の変更も可能です。

○イミディエイト・ウィンドウ

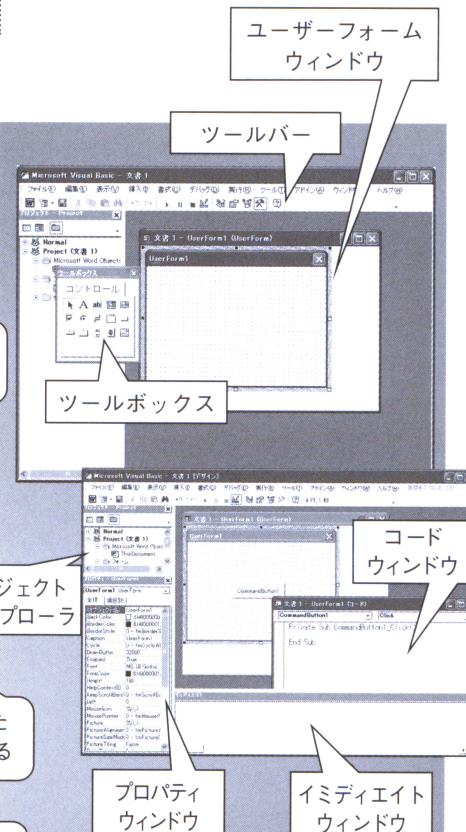
コードを入力して実行することができます。簡易的なテストやコードの確認に利用できます。

ツールバーを活用する

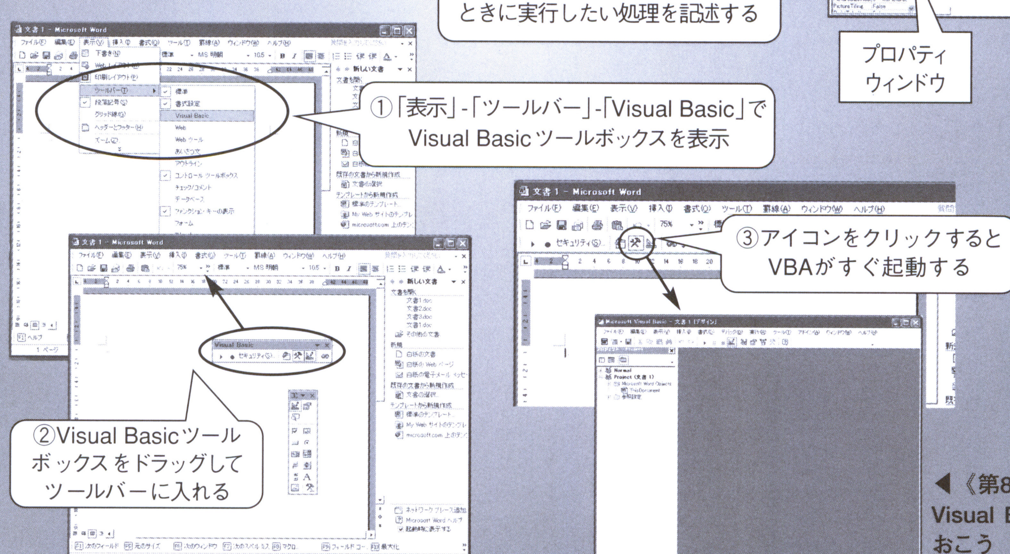
Visual Basicの編集画面を出す場合，メニューからいちいち選ぶのは面倒です。ツールバーにVisual Basicを登録しておくと，すぐにプログラムの編集を行うことができます。ツールバーに設定した内容は保存されますので，一度設定しておくことをお勧めします(**第8図**)。



▲《第6図》イベントごとに実行したい処理を記述する



▲《第7図》いろいろな機能のウィンドウがある



◀《第8図》ツールバーにVisual Basicを登録しておこう

教科「情報」など授業にも使える！

Visual Basic アルゴリズム講座

文・プログラム／東 幸太

第3回 加速度の計算をしてみよう

電車や自動車などに乗っていると，発進した時の加速や，ブレーキをかけた時の減速の様子が体感できます。このような，速度の変化がどの程度の時間で発生しているかを「加速度」という数値で表現します。今回は，この加速度をVisual Basicで計算してみます。

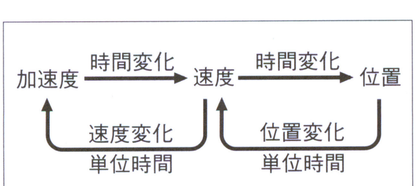


年度の区切りは年の区切りと異なり,パソコン的には何も影響はない。けれど,心機一転を図るには良いチャンスと言える。新しい環境になって心機一転を図っている人が多いと,環境が変わらない筆者も何か新しいことを始めようか,と思えてくる。

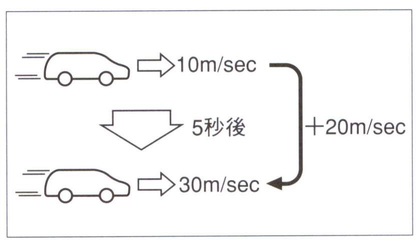
加速度とは

電車や自動車，エレベータなどの移動物体に人間が乗ると，物体の速度が変化するときに，人間も加速や減速の様子を体感することができます。物体の速度は瞬間的に変化することはなく，程度の差はあるものの連続的に変化していきますが，この変化の様子が加速感や減速感として感じとれるわけです。物体の位置が瞬間的に変化することはなく，必ず連続的に移動していくため，移動の様子が見られるのと同じ関係です。

この加速感や減速感を数値として表現するのが加速度です。単位時間当たりの位置の変化は速度に対応しますが，加速度はこの速度の単位時間当たりの変化量に対応します。加速度によって，速度が時間経過と共にどのように変化していくかが決まります。加速度がプラスの値であれば，速度は時間と共にプラス方向に変化していきますし，マイナスであれば速度はマイナス方向に変化していきます。これも，速度がプラス方向ならば位置もプラス方向に変化していき，逆にマイナス方向であればマイナス方向に変化するのと同様の関係です。位置，速度，加速度は**第1図**のような関係になっているわけです。



《第1図》位置・速度・加速度の関係

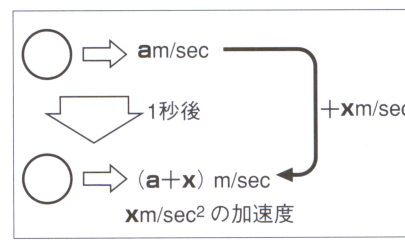


《第2図》加速する自動車

第2図のように10m/secの速度だった自動車が，５秒後に30m/secまで速度を上げた場合，速度の変化は+20m/secです。変化にかかった時間は５秒なので，

$$\frac{20m/sec}{5sec} = 4m/sec^2$$

となり，１秒当たり4m/secの速度変化があったことになります。１秒当たりｘm/secの速度変化がある場合，加速度がｘm/sec²であると表現します（**第3図**）。よって，この場合は加速度が4m/sec²となります。



《第3図》加速度の表現方法

地球上には重力が働いている関係で，物体は下の方に向かって落ちていきますが，この時には一定速度で落下するのではなく，一定の加速度で落下速度を上げていきます。この加速度のことを重力加速度と呼んでいて，9.8m/sec²程度の値になります。空気抵抗などを無視した場合，０速度から落下を始めて５秒後の物体は，

$$9.8m/sec^2 \times 5sec = 49m/sec$$

となり，秒速49mの速度に達していることになります。

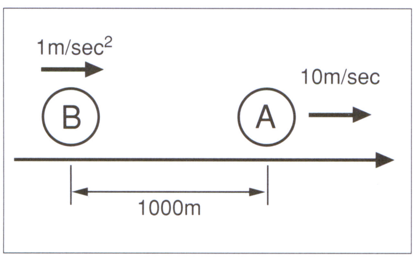
加速度による速度変化

加速度は速度の変化量なので，加速度が一定ならば，速度は一定のペースで変化していきます。その速度によって位置が変化していくことから，加速度が最終的な移動距離に与える影響は非常に大きくなります。

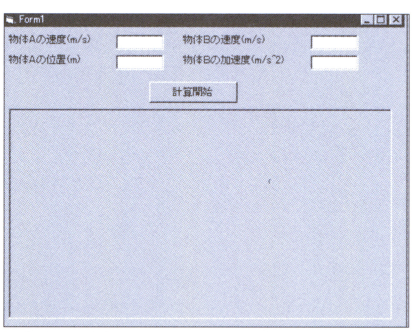
例えば，**第4図**のように秒速10mの一定速度で進んでいる物体Ａがあり，その

1000m後方に停止している物体Bに1m/sec²の加速度を与えて物体Aを追いかけさせるとどうなるでしょうか。10秒経過するまでは，BよりもAの速度が速いため，二つの物体間の距離は離れるばかりです。しかし，10秒経過すると二つの物体の速度は同じになり，それ以後はBが急速にAを追い上げる形になってきます。

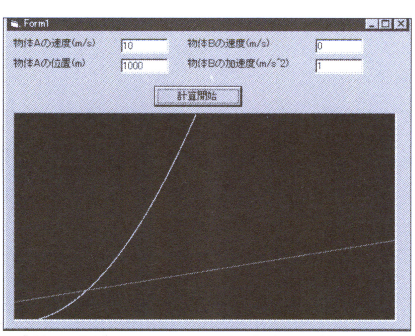
リスト1はこのような状況を計算し，二つの物体の位置と速度を画面表示するプログラムです。実行すると**第5図**のような画面が表示されます。物体Aの速度と最初の位置，そして物体Bの最初の速度と加速度を入力し，計算開始ボタンを押してください。横軸時間，縦軸位置のグラフの形で，二つの物体の移動の様子を表示します。なお，物体Bの最初の位置は0で固定です。



《第4図》一定速度のAを加速するBが追いかける



《第5図》リスト1を実行した直後の画面



《第6図》リスト1を実行し第4図の条件を入力した結果

第4図のような状況を入力すると，**第6図**のような結果が表示されます。物体Aは一定速度で移動するため，グラフでの移動の様子は直線状になります。これに対して物体Bは，速度が一定のペースで速くなっていくため，グラフでの移動

《リスト1》位置と速度を表示するプログラム・リスト

```
VERSION 5.00
Begin VB.Form Form1
  Caption         =  "Form1"
  ClientHeight    =  5292
  ClientLeft      =  48
  ClientTop       =  324
  ClientWidth     =  7200
  LinkTopic       =  "Form1"
  ScaleHeight     =  5292
  ScaleWidth      =  7200
  StartUpPosition =  3  'Windows の既定値
  Begin VB.CommandButton Command1
    Caption        =  "計算開始"
    Height         =  372
    Left           =  2640
    TabIndex       =  9
    Top            =  960
    Width          =  1572
  End
  Begin VB.TextBox Text4
    Height         =  264
    Left           =  5520
    TabIndex       =  8
    Top            =  480
    Width          =  852
  End
  Begin VB.TextBox Text3
    Height         =  264
    Left           =  5520
    TabIndex       =  7
    Top            =  120
    Width          =  852
  End
  Begin VB.TextBox Text2
    Height         =  264
    Left           =  2040
    TabIndex       =  4
    Top            =  480
    Width          =  852
  End
  Begin VB.TextBox Text1
    Height         =  264
    Left           =  2040
    TabIndex       =  3
    Top            =  120
    Width          =  852
  End
  Begin VB.PictureBox Picture1
    Height         =  3732
    Left           =  120
    ScaleHeight    =  3684
    ScaleWidth     =  6804
    TabIndex       =  0
    Top            =  1440
    Width          =  6852
  End
  Begin VB.Label Label4
    Caption        =  "物体Bの加速度(m/s^2)"
    Height         =  252
    Left           =  3240
    TabIndex       =  6
    Top            =  480
    Width          =  1932
  End
  Begin VB.Label Label3
    Caption        =  "物体Bの速度(m/s)"
    Height         =  252
    Left           =  3240
    TabIndex       =  5
    Top            =  120
    Width          =  1932
  End
  Begin VB.Label Label2
    Caption        =  "物体Aの位置(m)"
    Height         =  252
    Left           =  120
    TabIndex       =  2
    Top            =  480
    Width          =  1812
  End
  Begin VB.Label Label1
    Caption        =  "物体Aの速度(m/s)"
    Height         =  252
    Left           =  120
    TabIndex       =  1
    Top            =  120
    Width          =  1692
  End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Private Sub Command1_Click()
  Dim apos!, asp!, bpos!, bsp!, bacc!, ti!, timax!, dt!
  Dim xsize!, ysize!
  Dim f%

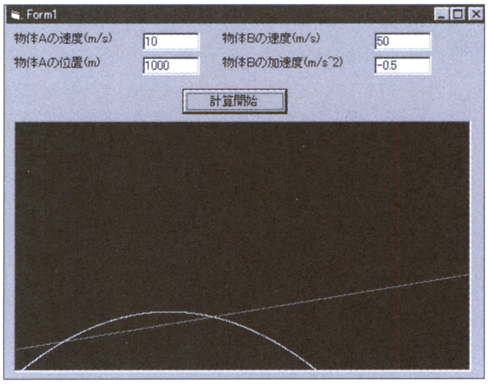
  asp = Text1.Text '物体Aの速度
  apos = Text2.Text '物体Aの位置
  bsp = Text3.Text '物体Bの速度
  bacc = Text4.Text '物体Bの加速度

  bpos = 0 '物体Bは0の位置からスタート

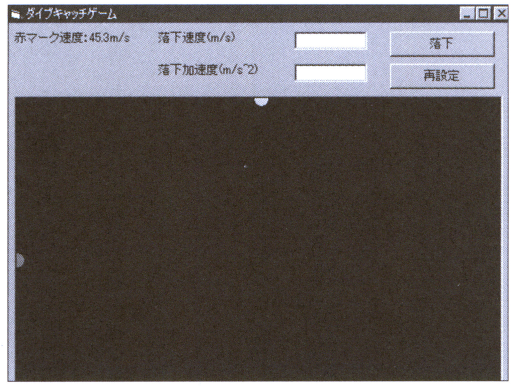
  dt = 0.05 '計算刻み時間

  xsize = Picture1.Width 'ピクチャボックスのサイズを取得
  ysize = Picture1.Height
  Picture1.Line (0, 0)-(xsize, ysize), 0, BF
  ti = 0 '時間
  timax = 300 '計算期間
  f = 0 '計算継続フラグ
  While f = 0
    apos = apos + asp * dt '速度に応じて位置を変化させていく
    bsp = bsp + bacc * dt '加速度に応じて速度を変化させていく
    bpos = bpos + bsp * dt
    Picture1.PSet (ti / timax * xsize, ysize - apos / 10000 * ysize), RGB(255, 0, 0) '物体Aの位置のプロット
    Picture1.PSet (ti / timax * xsize, ysize - bpos / 10000 * ysize), RGB(0, 255, 0) '物体Bの位置のプロット
    ti = ti + dt '時間を進める
    If ti > timax Then f = 1 '計算期間を越えたら終了
  Wend
End Sub

Private Sub Form_Load()
  Command1.Caption = "計算開始" 'キャプションの設定
  Label1.Caption = "物体Aの速度(m/s)"
  Label2.Caption = "物体Aの位置(m)"
  Label3.Caption = "物体Bの速度(m/s)"
  Label4.Caption = "物体Bの加速度(m/s^2)"
  Text1.Text = "" 'テキストボックスのクリア
  Text2.Text = ""
  Text3.Text = ""
  Text4.Text = ""
End Sub
```



《第7図》リスト1で物体Bの初期速度を50m/sec，加速度を－0.5m/sec^2とした場合の結果



《第8図》リスト2を実行した直後の画面

の様子は曲線となり，時間がたつにつれて速度が上がり，移動のペースが速くなっていくことが判ります。

次に，物体Bの加速度がマイナスの値になっている場合を見てみましょう。物体の速度と加速度の符号が異なる場合，その物体にはブレーキがかかっているかたちになり，速度の絶対値は徐々に小さくなっていきます。そして，速度が0になった後は，元と逆符号の速度になっていくため，物体は後戻りを始めます。

リスト1では物体Bの初期速度を設定できるようになっています。物体Bの初期速度を50m/sec，加速度を－0.5m/sec^2としてやると，**第7図**のような結果が表示されます。速度がプラスになっている間は前に進んでいますが，加速度がマイナスになっているため速度はどんどん低下していき，最後には0になります。その後，速度もマイナスになっていくため，物体Bは後戻りを始め，逆方向へどんどん速度を上げていきます。この時に物体Bの軌跡はなめらかなカーブになりますが，このカーブを放物線と呼びます。物体を上に投げた時には最初の速度は上方向ですが，重力加速度は常に下方向に加わるため，このようなカーブになることから「放物」という名前になっています。物体の速度と加速度の符号が逆になっていると，このようなカーブになります。

ダイブキャッチゲーム

それでは加速度をテーマにしたゲームをプレイしてみましょう。プログラムは**リスト2**です。

リスト2を実行すると，**第8図**のような画面が表示されます。画面上部には緑のマークがあり，画面左には赤いマークがあります。落下速度と落下加速度を指定して落下ボタンを押すと，赤いマークは画面に表示される一定速度で左から右へ移動を始め，緑のマークは指定された速度と加速度に応じて落下を始めます。ちょうど，緑のマークが赤いマークに当たるように速度と加速度を調整してみてください。再設定ボタンを押すと，赤いマークの速度が変わります。

マークが当たる場所までの距離は赤いマークと緑のマークとで異なるため，赤いマークの速度を参考にして調整した速度を設定する必要があります。とはいえ，速度だけを調整するのであれば比較的簡単です。しかし，速度を0として加速度だけで当てようとすると，後半の速度がかなり速くなるため難しくなります。

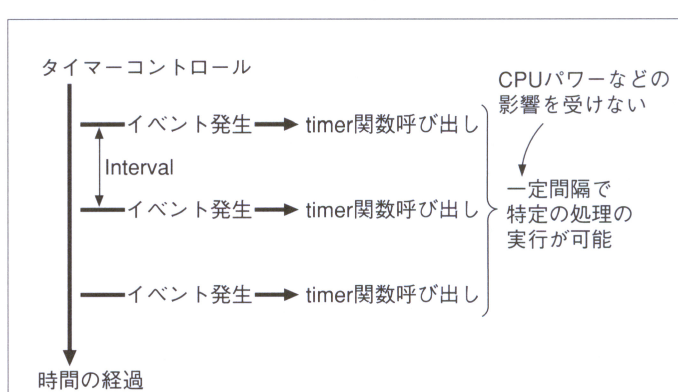
リスト2ではタイマーコントロールを使って，マークの移動の様子を表示しています。タイマーコントロールは，他のコントロールと違い，ウィンドウ上には影響を及ぼしません。設定した間隔でイベントを発生させ，イベントに対応した特定のサブルーチンを実行するという内部動作を行います。この特定の関数がtimerになります（**第9図**）。

タイマーコントロールのプロパティとしては，イベント発生間隔を指定するIntervalと，タイマーコントロールの動作のOn/Offを指定するEnabledがあります。Intervalにはミリ秒単位（1/1000秒単位）でイベント発生間隔を指定します。100を設定すると，0.1秒に1回イベントが発生するわけです。Enabledはイベントを発生させるかどうかを指定します。

パソコンの処理能力が異なると，通常はプログラムの実行速度が異なってくるのですが，タイマーコントロールを使用すれば実際の時間によるプログラムコントロールができるようになります。

ダイブキャッチゲームでは，マークが移動する様子を表示する必要があります。マークを単純に移動させてしまうと，処理能力の高いパソコンでは，あっという間に移動が完了してしまいます。そこで，タイマーコントロールTimer1を用意し，Timer1_Timer関数で1回の呼び出しについて変数dtで指定された時間が進むと想定した移動処理を行っています。これによって，処理能力が異なっても1回のイベント発生についてdt分だけの内部時間が進むようになります。イベント発生はどのパソコンでも一定ですから，どのパソコンでも内部時間の進み方が同じになり，移動の様子の表示がパソコンの処理能力の影響を受けなくなるわけです。

なお，イベント発生間隔をあまり短くしてしまうと，イベント処理が終わらないうちに次のイベントが発生するようになります。こうなると，処理速度は落ちてしまい，期待した動きにはなりません。



《第9図》タイマーコントロールの動作

イベント発生間隔が短すぎる場合や，1回のイベントでの処理内容が多い場合はこの点に注意してください。

ダイブキャッチゲームはゲームとはいえ，赤いマークに緑のマークが当たったかどうかは判断していません。この処理を付け加え，どこまでマーク同士が接近したかを表示することを考えてみてください。マーク移動時に赤いマークと緑のマークの間の距離を計算し，その距離の最小値を保存するという方法が簡単です。そして，移動終了時に最小値を表示するようにすれば，それをそのままゲームのスコアとして使うことができます。各自で**リスト2**を改造してみてください。

おわりに

加速度は速度に対して影響を与えるため，位置に対する影響は時間の経過と共に大きくなっていきます。いろいろな条件での動きを見て，加速度の感覚をつかんでください。

《リスト2》加速度をテーマにした「ダイブキャッチゲーム」のプログラム・リスト

```
VERSION 5.00
Begin VB.Form Form1
   Caption         =   "Form1"
   ClientHeight    =   6012
   ClientLeft      =   48
   ClientTop       =   324
   ClientWidth     =   7584
   LinkTopic       =   "Form1"
   ScaleHeight     =   6012
   ScaleWidth      =   7584
   StartUpPosition =   3  'Windows の既定値
   Begin VB.CommandButton Command2
      Caption         =   "Command2"
      Height          =   372
      Left            =   5760
      TabIndex        =   7
      Top             =   600
      Width           =   1572
   End
   Begin VB.Timer Timer1
      Left            =   120
      Top             =   600
   End
   Begin VB.CommandButton Command1
      Caption         =   "Command1"
      Height          =   372
      Left            =   5760
      TabIndex        =   6
      Top             =   120
      Width           =   1572
   End
   Begin VB.TextBox Text2
      Height          =   264
      Left            =   4320
      TabIndex        =   4
      Text            =   "Text2"
      Top             =   600
      Width           =   1092
   End
   Begin VB.TextBox Text1
      Height          =   264
      Left            =   4320
      TabIndex        =   3
      Text            =   "Text1"
      Top             =   120
      Width           =   1092
   End
   Begin VB.PictureBox Picture1
      Height          =   4812
      Left            =   120
      ScaleHeight     =   4764
      ScaleWidth      =   7284
      TabIndex        =   0
      Top             =   1080
      Width           =   7332
   End
   Begin VB.Label Label3
      Caption         =   "Label3"
      Height          =   252
      Left            =   2280
      TabIndex        =   5
      Top             =   600
      Width           =   1932
   End
   Begin VB.Label Label2
      Caption         =   "Label2"
      Height          =   252
      Left            =   2280
      TabIndex        =   2
      Top             =   120
      Width           =   1932
   End
   Begin VB.Label Label1
      Caption         =   "Label1"
      Height          =   252
      Left            =   120
      TabIndex        =   1
      Top             =   120
      Width           =   1812
   End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Dim gs% 'ゲームシーケンス状況
Dim redx!, redy!, redv! '赤マークの座標と速度
Dim greenx!, greeny!, greenv!, greena! '緑マークの座標・速度・加速度
Dim xsize!, ysize! 'ピクチャボックスのサイズ
Dim dt! '刻み時間
Dim marksize! 'マークのサイズ

Private Sub Command1_Click()
   Picture1.Line (0, 0)-(xsize, ysize), 0, BF 'ピクチャボックス背景を黒で塗りつぶす
   redx = 0 '赤マーク位置設定
   greeny = 0 '緑マーク位置設定
   greenv = Val(Text1.Text) '緑マークの速度
   greena = Val(Text2.Text) '緑マークの加速度
   gs = 2 'シミュレーションスタート
End Sub

Private Sub Command2_Click()
   gs = 0
End Sub

Private Sub Form_Load()
   Command1.Caption = "落下" 'キャプションの設定
   Command2.Caption = "再設定"
   Form1.Caption = "ダイブキャッチゲーム"
   Label1.Caption = ""
   Label2.Caption = "落下速度(m/s)"
   Label3.Caption = "落下加速度(m/s^2)"
   Text1.Text = "" 'テキストボックスのクリア
   Text2.Text = ""
   dt = 0.5 '刻み時間
   gs = 0 'ゲームシーケンス
   xsize = Picture1.Width 'ピクチャボックスのサイズを取得
   ysize = Picture1.Height
   Picture1.FillStyle = vbFSSolid
   marksize = 100 'マークのサイズ
   Timer1.Interval = 20 'インターバルタイマ起動
End Sub

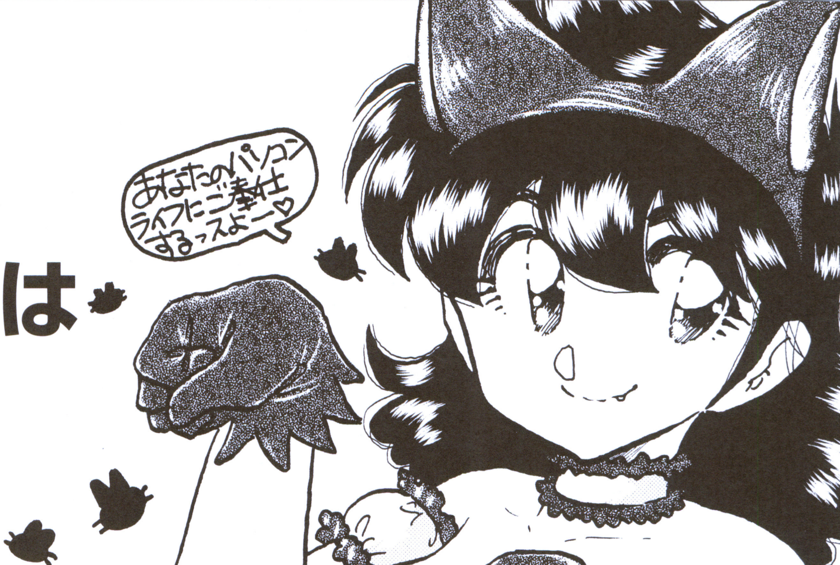
Private Sub Timer1_Timer()
   Dim newredx!, newgreeny!
   If gs = 0 Then '状況の作成
      Picture1.Line (0, 0)-(xsize, ysize), 0, BF 'ピクチャボックス背景を黒で塗りつぶす
      redx = 0 '赤マーク位置設定
      redy = ysize / 2
      redv = Rnd(1) * 50 + 10 '赤マーク移動速度設定
      Label1.Caption = "赤マーク速度：" + Format$(redv, "###.0m/s")
      greenx = xsize / 2 '緑マーク位置設定
      greeny = 0
      Picture1.FillColor = RGB(255, 0, 0) '赤マーク表示
      Picture1.Circle (redx, redy), marksize, RGB(255, 0, 0)
      Picture1.FillColor = RGB(0, 255, 0) '緑マーク表示
      Picture1.Circle (greenx, greeny), marksize, RGB(0, 255, 0)
      gs = 1
   ElseIf gs = 2 Then 'シミュレーション
      newredx = redx + redv * dt '赤マーク移動
      greenv = greenv + greena * dt '緑マーク加速
      newgreeny = greeny + greenv * dt '緑マーク移動
      Picture1.FillColor = 0 '赤マーク消去
      Picture1.Circle (redx, redy), marksize, 0
      Picture1.FillColor = RGB(255, 0, 0) '新しい位置に赤マーク表示
      Picture1.Circle (newredx, redy), marksize, RGB(255, 0, 0)
      redx = newredx
      Picture1.FillColor = 0 '緑マーク消去
      Picture1.Circle (greenx, greeny), marksize, 0
      Picture1.FillColor = RGB(0, 255, 0) '新しい位置に緑マーク表示
      Picture1.Circle (greenx, newgreeny), marksize, RGB(0, 255, 0)
      greeny = newgreeny
      If greeny > ysize Or redx > xsize Then gs = 3 'どちらかが画面をはみ出したら終了
   End If
End Sub
```

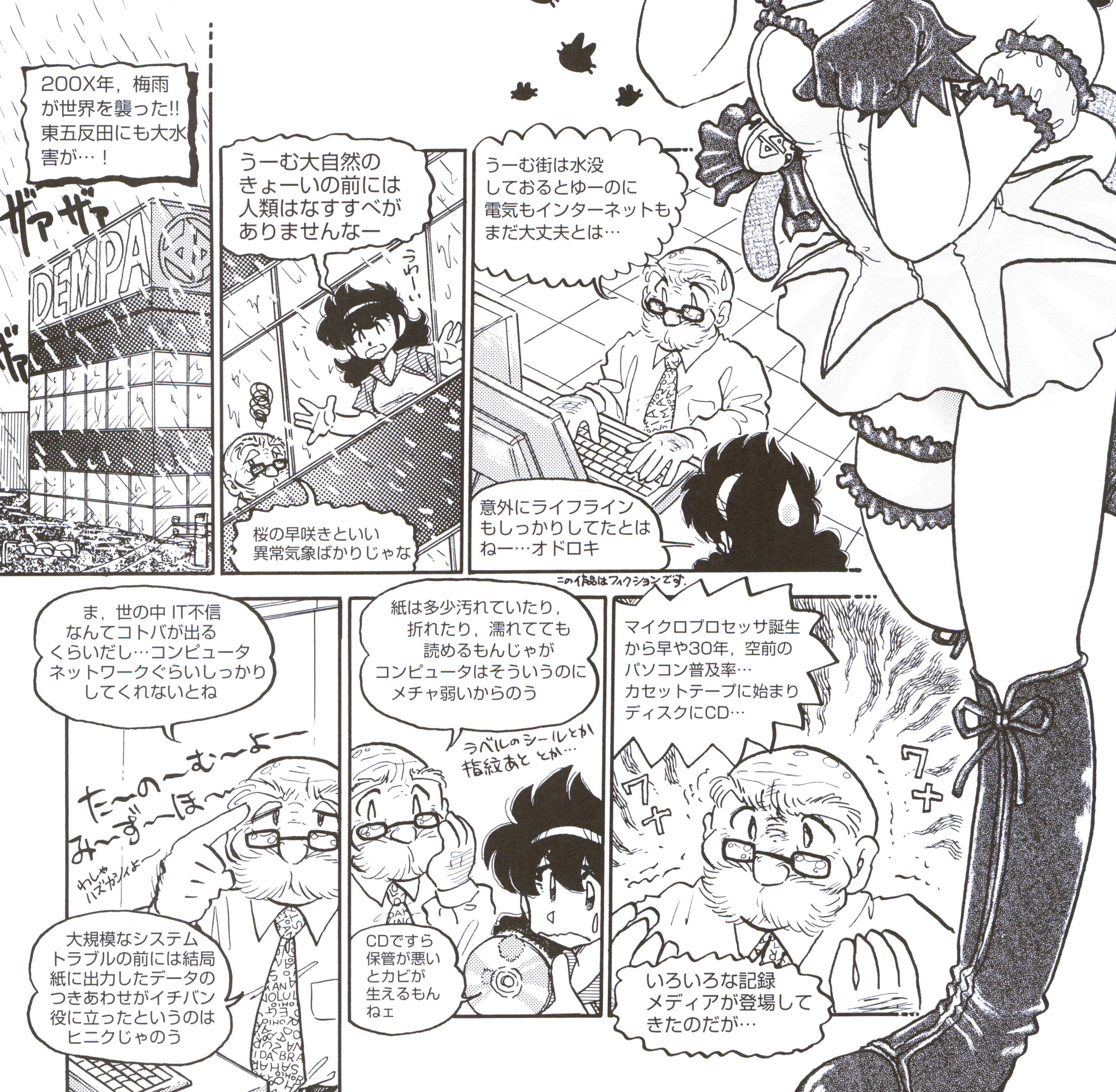
マンガで学ぶ **コンピュータの世界**

パソコンレクチャー ザ・ビギニング

「プリントアウトは百年持つ!?」

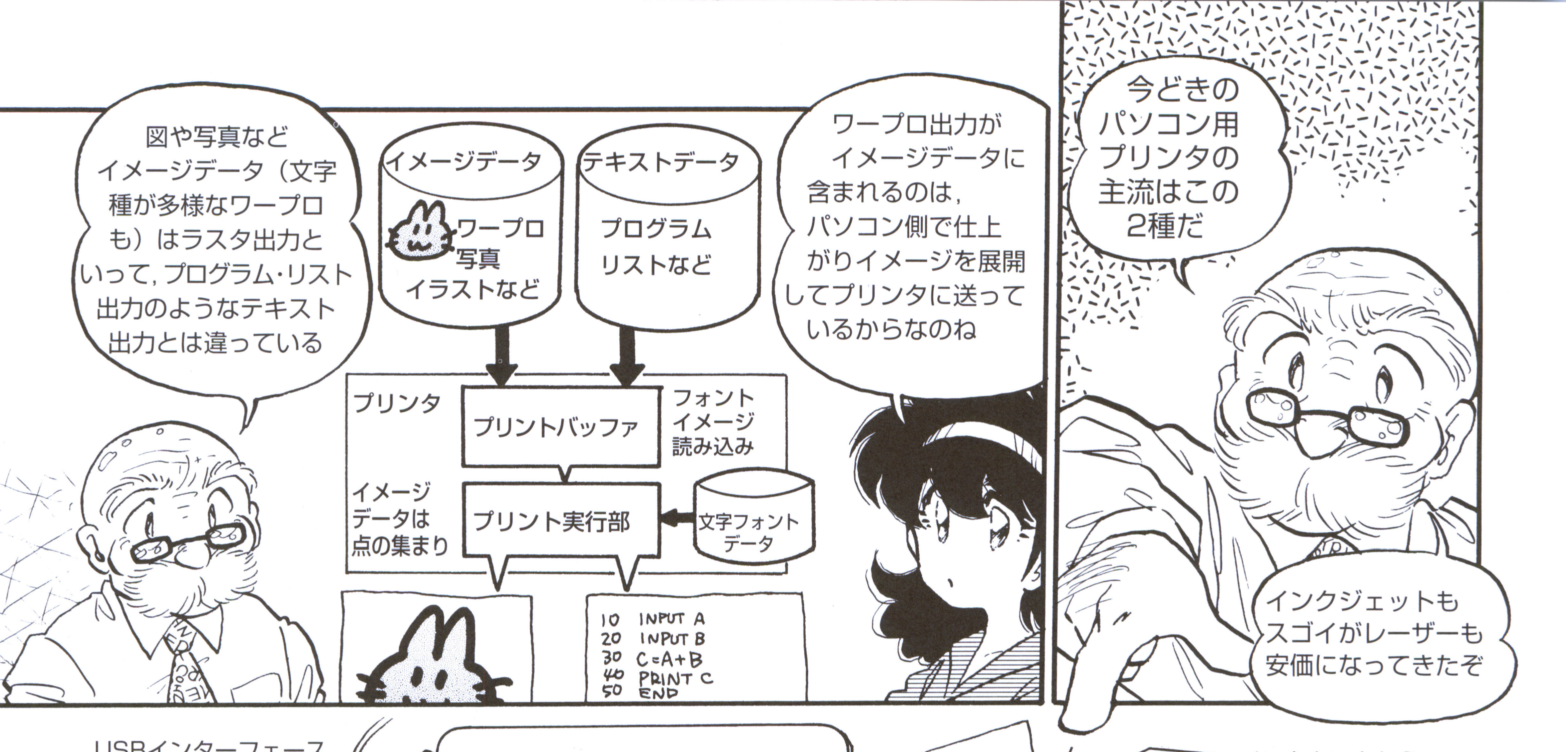
by くりひろし



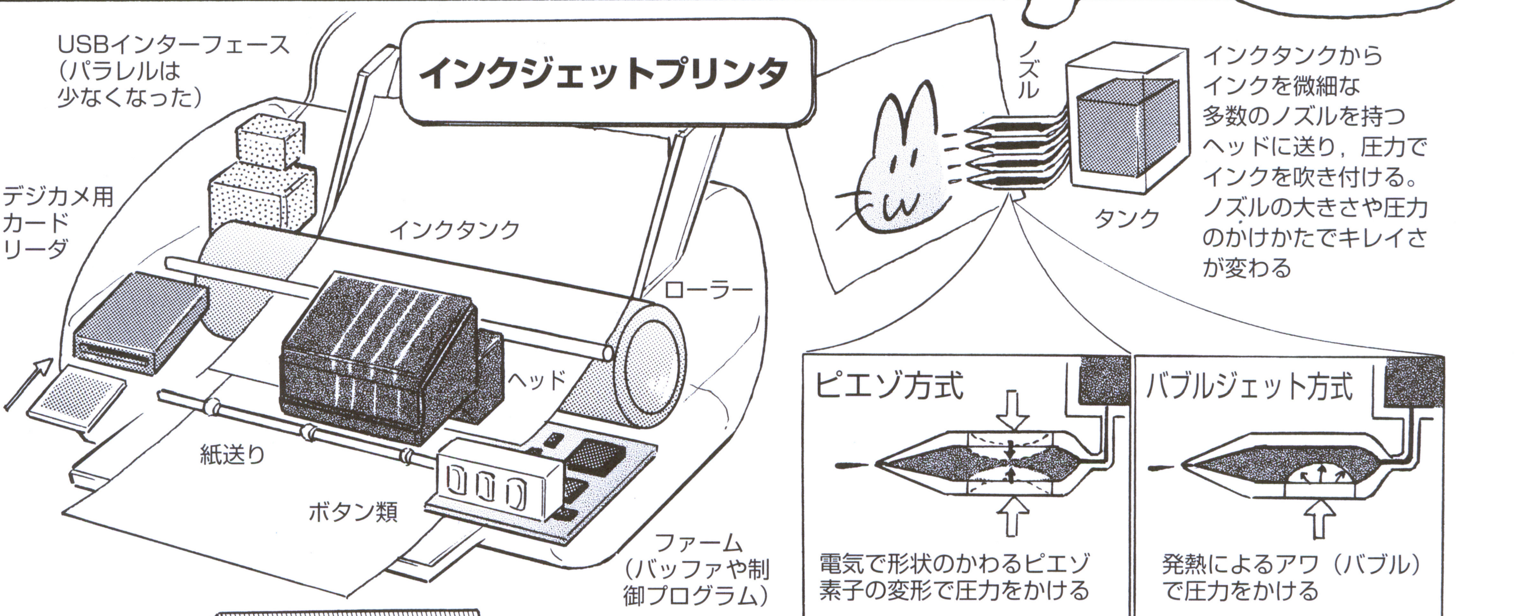




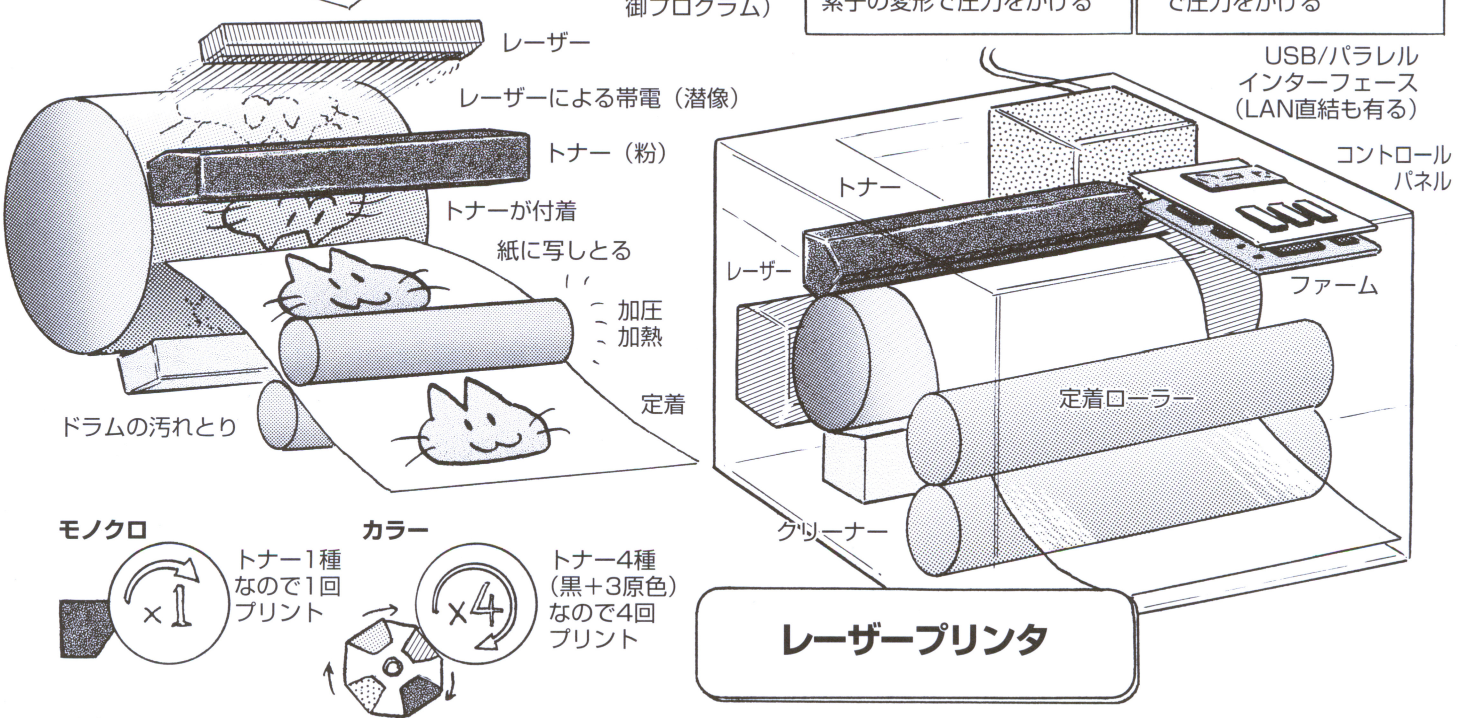
いまだ人類は
3千年前の紙の発明を
超えるメディアを生み
出せておらんのだぁぁっ
紙がナンボの
もんじゃーい
ちょっと火ィ付け
ればたちまち…
燃や
すなぁぁ
ディスクやMO,CDだって
火燃えるわー!!
メラ
メラ
ボッ
どんなハイテクメディア
だって，火にはもちろん
水にも弱い！
こーゆーのは
ふつー影くんの
ネタじゃのう
ちょっとやってみました。
<読者しょくんは絶対マネしないように!!>
火あぶり
フロッピー……×
CD……×
MO……×
水責め
フロッピー……×
CD……○
MO……×
HD……×
情報の耐久性においては，
紙にまさるメディアはないのじゃ!!
もちろん「人間の目」というデバイスに
よるところも大きいのだが…
と！いうわけで，
今回は「紙」に出力できる
コンピュータの重要な周辺
機器!!
プリンタのテスト
たぶんちゃんとうごいてるよ
ドライバーは
プリンタも
見てみよう!!
またゴーインな展開だなぁ
パソコンのセットでは
プリンタなんてオプション
なので軽く見られる事も
多いが…コンピュータの
生い立ちを考えると
プリンタの方が早くから
プログラミングにも役に
立ってきたのだ
ずっと昔のコンピュータでは，ディスプレイ
が標準的になるまでは文字のプリントアウト
ができるタイプライタ端末が一般的だったのだ
出力
入力
ダ
ダ
ダダ
1980
S.H.A.D.O
S.I.D
UNIXなどでは最小限の
入出力インターフェース
（標準入出力，文字の入力/
出力のみ）をttyと呼ぶ
のはこの名残じゃな
LOGIN: KURI
PASSWORD: ****
KURI LOGGED IN XXX.XX.XX
WHO AM I
> KURI TTY01
>
あまり一般の
読者諸君には
縁が無いかな？
でもUNIXだけ
でなくてLINUXとかも
同じ考え方じゃよ
現在のプリンタは文字しか出力できない
ということがなくなり，WYSIWYG
（What you see is what you get：見た
ものが得られるもの）が基本の，イメージ
プリンタに進化している
CPU
メモリ
入出力
制御
キー
ボード
ディス
プレイ
プリンタ
（イメージプリンタ）
アプリケーション
ソフトのプリント出力
・ワープロ文書
・CGイラスト
・デジカメ写真データ
・ホームページ
・表計算シート
・ポストカード
・プログラム・リスト



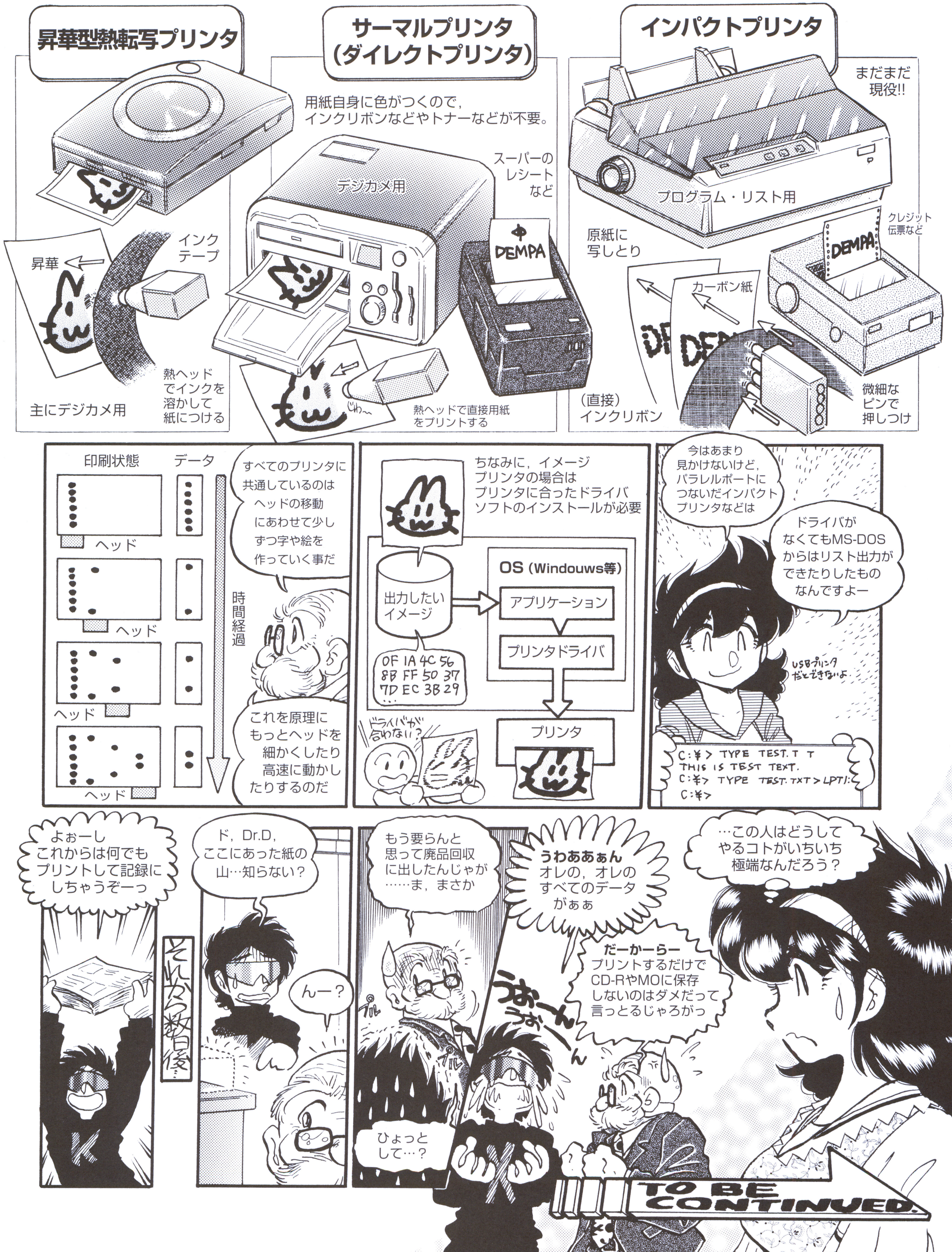
図や写真など
イメージデータ（文字種が多様なワープロも）はラスタ出力といって，プログラム・リスト出力のようなテキスト出力とは違っている
イメージデータ
ワープロ
写真
イラストなど
テキストデータ
プログラム
リストなど
プリンタ
プリントバッファ
フォント
イメージ
読み込み
イメージ
データは
点の集まり
プリント実行部
文字フォント
データ
10 INPUT A
20 INPUT B
30 C=A+B
40 PRINT C
50 END
ワープロ出力がイメージデータに含まれるのは，パソコン側で仕上がりイメージを展開してプリンタに送っているからなのね
今どきのパソコン用プリンタの主流はこの2種だ
インクジェットもスゴイがレーザーも安価になってきたぞ



インクジェットプリンタ
USBインターフェース
（パラレルは
少なくなった）
デジカメ用
カード
リーダ
インクタンク
ローラー
ヘッド
紙送り
ボタン類
ファーム
（バッファや制
御プログラム）
ノズル
タンク
インクタンクから
インクを微細な
多数のノズルを持つ
ヘッドに送り，圧力で
インクを吹き付ける。
ノズルの大きさや圧力
のかけかたでキレイさ
が変わる
ピエゾ方式
電気で形状のかわるピエゾ
素子の変形で圧力をかける
バブルジェット方式
発熱によるアワ（バブル）
で圧力をかける



レーザー
レーザーによる帯電（潜像）
トナー（粉）
トナーが付着
紙に写しとる
加圧
加熱
定着
ドラムの汚れとり
モノクロ
×1
トナー1種
なので1回
プリント
カラー
×4
トナー4種
（黒＋3原色）
なので4回
プリント
USB/パラレル
インターフェース
（LAN直結も有る）
コントロール
パネル
トナー
レーザー
ファーム
定着ローラー
クリーナー
レーザープリンタ



昇華型熱転写プリンタ
主にデジカメ用
昇華
インクテープ
熱ヘッドでインクを溶かして紙につける
サーマルプリンタ（ダイレクトプリンタ）
用紙自身に色がつくので，インクリボンなどやトナーなどが不要。
デジカメ用
スーパーのレシートなど
DEMPA
熱ヘッドで直接用紙をプリントする
じゅわ〜
インパクトプリンタ
まだまだ現役!!
プログラム・リスト用
クレジット伝票など
DEMPA
原紙に写しとり
カーボン紙
（直接）インクリボン
微細なピンで押しつけ
印刷状態
データ
ヘッド
ヘッド
ヘッド
ヘッド
時間経過
すべてのプリンタに共通しているのはヘッドの移動にあわせて少しずつ字や絵を作っていく事だ
これを原理にもっとヘッドを細かくしたり高速に動かしたりするのだ
ちなみに，イメージプリンタの場合はプリンタに合ったドライバソフトのインストールが必要
出力したいイメージ
OS（Windouws等）
アプリケーション
プリンタドライバ
プリンタ
0F 1A 4C 56
8B FF 50 37
7D EC 3B 29
ドライバが合わない!?
今はあまり見かけないけど，パラレルポートにつないだインパクトプリンタなどは
ドライバがなくてもMS-DOSからはリスト出力ができたりしたものなんですよー
USBプリンタだとできないよ。
C:¥> TYPE TEST.T T
THIS IS TEST TEXT.
C:¥> TYPE TEST.TXT > LPT1:
C:¥>
よぉーし　これからは何でもプリントして記録にしちゃうぞーっ
それから数日後…
ド，Dr.D，ここにあった紙の山…知らない？
んー？
もう要らんと思って廃品回収に出したんじゃが……ま，まさか
ひょっとして…？
うわああぁん　オレの，オレのすべてのデータがぁぁ
うおーん　うおーん
だーかーらー　プリントするだけでCD-RやMOに保存しないのはダメだって言っとるじゃろがっ
…この人はどうしてやるコトがいちいち極端なんだろう？
TO BE CONTINUED.

■連載■ PCハードウエアテクニック

はじめてPC製作にチャレンジするための知識＆テクニックを徹底解説

DOS/Vマシン自作 完全マニュアル①

メーカーの既製マシンではなく，パーツを集めてオリジナル・マシンにチャレンジ！とうい人も，珍しくなくなりました。とくに，DTM（デスクトップ・ミュージック），最近話題のDTV（デスクトップ・ビデオ編集），サーバーなど，用途に特化した高性能マシンを導入する場合には，自作PCは最適です。さらに低予算でベース・マシンを入手し，少しずつアップグレードを行い，高性能なマシンに作り上げることができるのも，自作DOS／V機ならではの特徴でしょう。今回から魅力いっぱいの自作マシンに挑戦するための基礎知識や，さまざまな製作テクニックを紹介します。

1 自作マシン導入の基本「DOS／V機の構成」

国内外のメーカーから，多数のパソコンが発売されています。AV機能を強化したモデルやインターネット対応をウリにしている機種などさまざまですが，そのなかから，自分の用途に合った機種を見つけるとなると，大変な作業です。また，機能面では合格だが，処理性能が今ひとつという，「帯に短しタスキには長し」的な悩みも少なからずあります。

そこでオススメしたいのが，自作マシンです。メーカー独自の機能を加えたモデルもありますが，既製パソコンの仕様は，いずれもDOS／V規格となってます。DOS／Vマシンの特徴は，さまざまな周辺機器が利用できる拡張性の高さと，パーツの互換性に優れていることですが，この独自機能が壁になり，既製パソコンは，DOS／V機本来の拡張性や互換性に問題がある場合も多く，自分の用途にピッタリのパソコン環境を手に入れたい人には，物足りなさを感じてしまうこともしばしばです。

その点，パーツを自ら集めて1台のパソコンに仕上げる自作DOS／Vマシンであれば，パーツの選択次第で，目的に合ったマシンに仕上げることが可能です。「自作」というと，ハードウエアが苦手で，とても難しいのでは思われるかも知れませんね。ところが，パーツさえ揃えてしまえば，組み立て作業はドライバー1本だけで済んでしまうほど簡単で，パソコンが初めてという場合でも，基本的なルールさえ守れば，問題なく自分だけのオリジナル・パソコンを作ることができるのです。それに加えて，あとから処理性能をアップしたり，機能を追加する自由度のひろさも，自作DOS／V機ならではの特徴といえます。

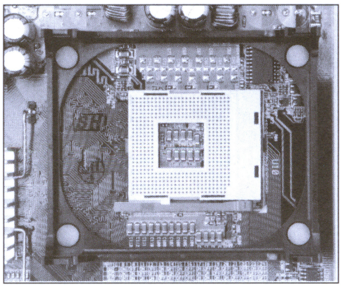
PC自作への第一歩はDOS／V機の仕組みをマスタ

ドライバー1本で済む自作DOS／V

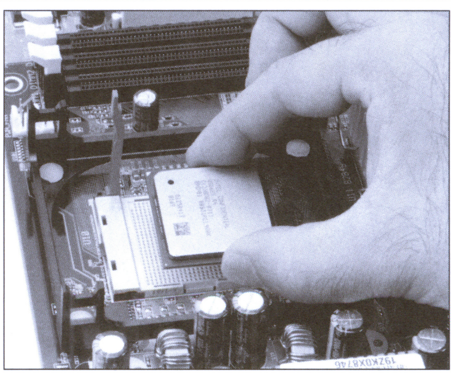
●最新情報／CPU編

DOS／Vパソコンの心臓部にあたるパーツといえば「CPU」。このCPUはいくつか規格がありますが，熾烈な高性能競争を繰り広げているのが，インテル社のペンティアム，セレロンとAMD社のアスロン，デュロンの各シリーズです。

インテル系では，最速のペンティアム4，AMD系はペンティアム4と同等の性能を誇るアスロンXPが，ハイエンドCPUとなっていますが，ソケットの互換性はなく，選択には各CPUごとに対応した専用のマザーボードが必要になります。とはいえ，処理性能を抑えたセレロンまたはデュロン・シリーズでも，一般的な用途には，1GHzオーバーのCPUなら，まったく問題なく導入できます。価格もグーンとお得なので，最初のマシンにはオススメです。



▶ペンティアム4のCPUソケット「マイクロPGA」（ソケット478）



▲ペンティアム4は現在，駆動周波数2.4GHzのタイプが最速

パソコンですが，手順どおりに組み立てるだけでは，せっかくの自作が台無しです。というのも，DOS／V機の各パーツの働きや構成をおおざっぱでも良いですから，知識として身につけておいたほうが，組み立て後のチューンナップを行う場合や，ほかの周辺機器との連携を行う場合に，とても役立つからです。

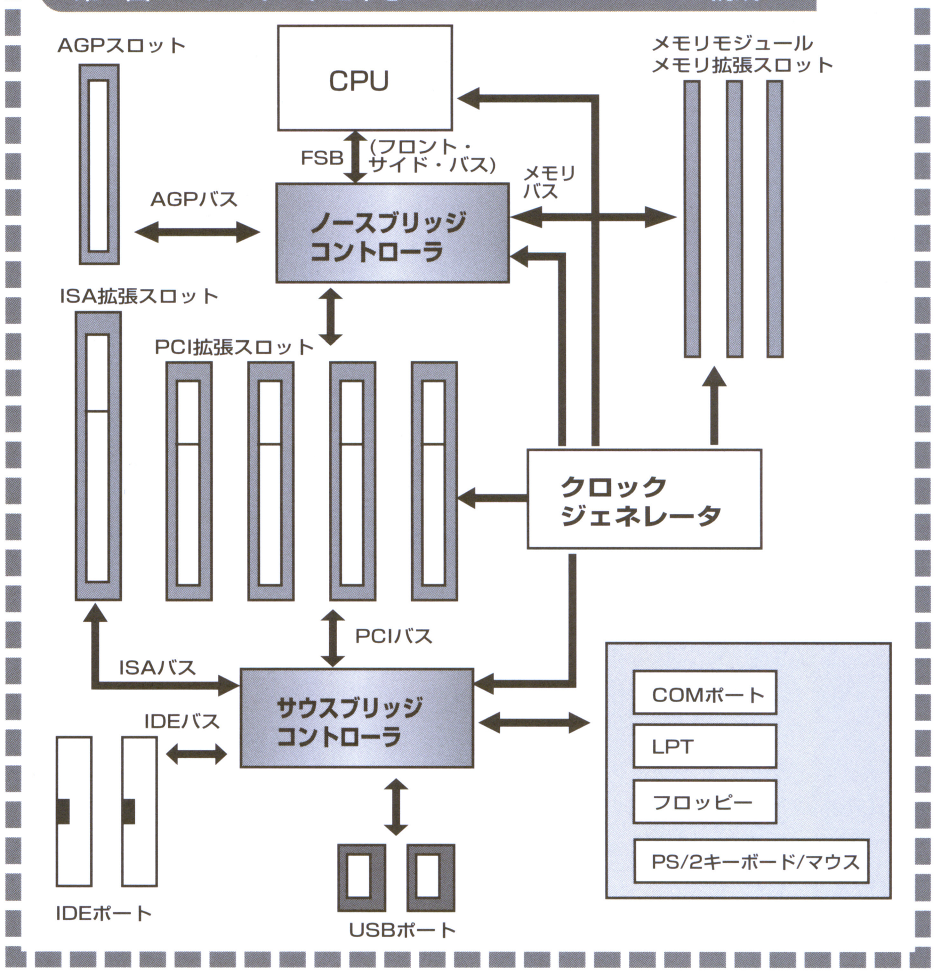
その一つとして重要な基本知識が，DOS／V機の核となる，マザーボードの構成とその働きです。**第１図**は，マザーボードの仕組みを示したブロック図です。

この図でマスターしておきたいのが，パソコンの重要な構成パーツである，CPU(中央演算ユニット)とメモリ，ハードディスク。さらに，さまざまな拡張ボードを増設することになるPCIスロットとの関係です。マザーボードの話題で頻繁に出てくる「チップセット」という用語がありますが，各構成パーツとのデータのやりとりは，このチップセットを介して行います。チップセットは，CPUとメモリと連携する「ノースブリッジ・コントローラ」とキーボードやプリンタ，USB接続の周辺機器，さらに内蔵HDD（ハードディスク）との連携を目的とした「サウスブリッジ・コントローラ」の二つの専用ＩＣの組み合わせとなっています。

導入できるＣＰＵの種類やメモリの規格だけでなく，パソコン全体の処理性能に直結するFSB（フロント・サイドバス）の動作周波数などは，サウスブリッジ・コントローラで，利用できる周辺機器との基本インターフェースの規格は，サウスブリッジ・コントローラの種類でほぼ決まってしまいます。

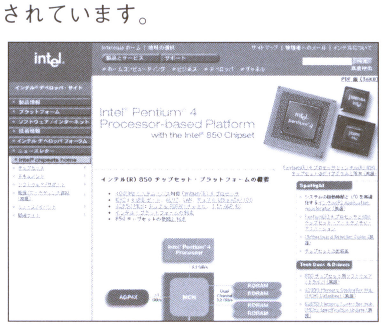
要するに，自作機に求める機能や性能はマザーボード次第，または採用しているチップセットの選択で決定されるとも言って良いわけです。マザーボードやチップセットの詳細については，機会をあらためて紹介するつもりですが，まず，**第１図**でパソコンの構成は，このようなものであるということだけは知っておいてください。

第１図　マザーボードを中心にしたDOS/Vパソコンの構成

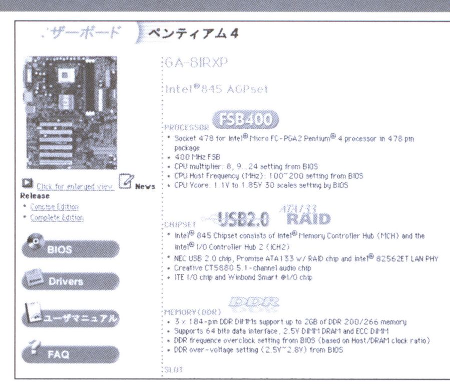


●最新情報／マザーボード編

CPUが心臓部なら，マザーボードはDOS/V機のベースにあたる重要なパーツです。マザーボードの性能を決定づけるチップセットの種類もまた，数多くのメーカーが扱っています。CPUメーカーであるインテル社もチップセットを製造しており，ペンティアム４の最新チップセット「850」は，高速なメモリ規格「RIMM」やHDDインターフェース規格「ウルトラATA/100」など，ハイエンド向けの仕様を持っています。さらに同社の「845」チップセットは，RIMMの対抗馬である「DDR」メモリ規格に対応しているのが特徴。このほか，HDD接続規格「ウルトラATA/100」にも対応。また，FSB533MHzペンティアム４正式対応の「SiS645DX」チップセット搭載マザーも発売されています。



▶お馴染みのインテル社も有力なチップセット・メーカーのひとつ



▲ハイエンド・マシンには最新のCPUやメモリ規格に合わせたマザーボードを選択

2 性能&機能を決定づける主要パーツはこれだ

さなざまなパーツで構成されるDOS／Vパソコンですが，なかでも中心となるのが，CPU，マザーボード，メモリ，ハードディスク（HDD）の四つ。ビデオカードやサウンドカードなども自作パーツには欠かせませんが，基本的な構成ということで省いています。

対応パーツを見極めてマザーボードを選択

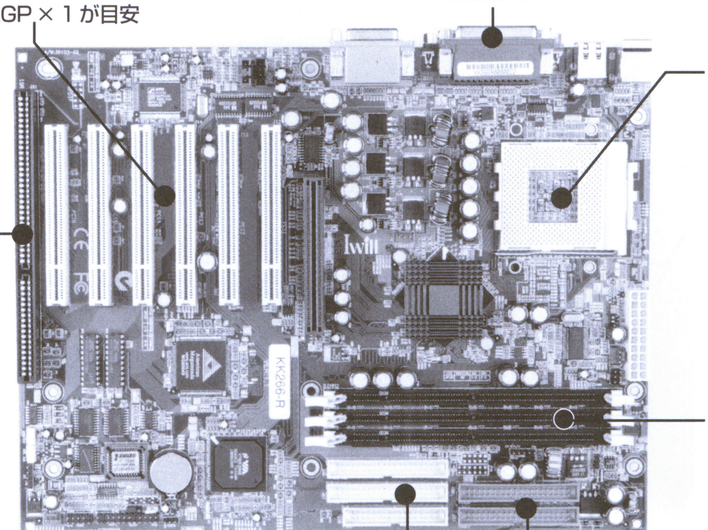
マザーボードは，すでに紹介したとおり，各パーツの中心となるもので，この選択で組み立て後のマシンの性能がほぼ決められてしまいます。さきほどの搭載チップセットの見極めももちろんですが，ボード上のソケットや端子（コネクタ），さらに搭載機能の事前チェックもベストなマザーボード選びのコツといってもかまいません。各チェック・ポイントについては，**第2図**を参考にしていただくとして，「低予算で最高の性能と機能を!」という，切実なニーズに応えられる選択方法を紹介しておきましょう。

マザーボードには，ハイエンド，ミドル・クラス，エントリー向けの三つに大別できます。ハイエンドは，余分な付加機能を省き，最速のCPUとメモリ，HDDの性能をフルに引き出せるように設計されたフルスペック・マザーです。これ以外の仕様は，すべてユーザーが用意する拡張ボードでカバーしなければならず，ある程度の予算（マザーボードは3万円以上）は覚悟しなければなりません。

エントリー向けは，CPU，メモリ以外はオールインワン，すなわち搭載済みのモデルで，低予算向け（数千円からある）ともいえます。しかし，拡張性（PCIスロット数が少ない），組み合わせ可能なCPUが限定されているなどの欠点が見られます。

両方の良いとこ取りができるのが，ミドルレンジ・クラスです。1～2万円の製品がこれにあたるわけですが，拡張スロットも四つ以上搭載し，高クロックのペンティアム4やアスロンXPにも対応可能です。サウンド機能も搭載し，基本的な用途であれば拡張ボードを別に用意しなくても済むだけでなく，種類も極めて豊富です。はじめてDOS／V機の自作にチャレンジする人も，ミドルクラスのマザーボードを選んでおけば，後のアップグレードにも安心です。マザー選択の決め手となる搭載チップセットは，

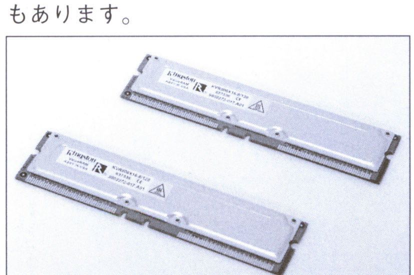
《第2図》マザーボードのチェック・ポイント



●最新情報／メモリ編

CPUとともにパソコンの処理性能の決め手になるメモリの規格は，SDRAM，DDR，RIMMの3種類ありスロット形状などの互換性はありません。CPUとメモリは連動してデータ処理を行うため，高速なCPUでもメモリの読み書き速度が遅ければ，本来の性能は引き出せません。そのため，最新規格であるDDR，RIMMメモリは，SDRAMの数倍の転送速度を持っています。

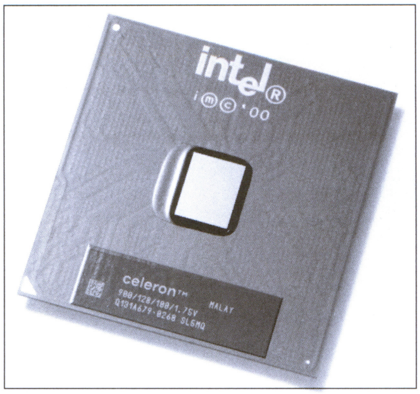
ただし，RIMMは高価なことから，DDRメモリが主流になりつつあります。同じ規格でも読み書き速度の違いで，いくつかの種類があり，導入するCPUやマザーボードに適合したメモリを選ばなくてはなりませんし，メモリ購入時の注意点でもあります。



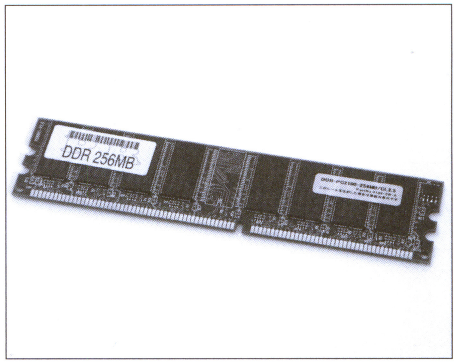
▲ペンティアムCPU向けとして登場したRIMMだが，高価で人気は今ひとつ



▲主流になりつつあるDDRメモリ。同クロックでもSDRAMの2倍の転送速度を持つ



▲ほとんどのコンピュータ処理を行うCPUの選択は慎重に行おう



▲CPUと連携してデータの読み書きを行うメモリは，できるだけ高速で大容量が基本。DDR規格が主流になりつつある



▲高速&大容量が求められるのは，メモリと同様。低価格化のおかげで，極めて安く高性能なHHDを手に入れられる

個人的な意見ですが，最新のものでなくても良いでしょう。安定したパソコン環境を目指す場合は，最新チップによく見られるバグや周辺パーツとの相性問題が逆に足かせになることもあります。

1GHz前後クラスの最新CPUがお買い得

高速であればあるほどベストなCPU選択ですが，駆動周波数が上がるほど高くつきます。最速のペンティアム4 2.4GHzは7万円以上もしますし，この価格であれば，DOS／Vマシンを丸ごと購入することも可能です。

価格と性能のバランスを考慮すると，1.6GHz前後のペンティアム4または，アスロンXP 1800+ (1.53GHz) がオススメです。ペンティアムⅢやセレロン，デュロンなども，予算が足りない場合には選択の余地はあるものの，この先数年同じマシンで済ましたい場合では，少々無理をしてもペンティアム4かアスロンXPをチョイスしておいたほうが良いでしょう。ペンティアムⅢから4への乗り換えは，マザーボードの新規購入をともなうので，アップグレードを前提にするには，最初から最新バージョンのCPUの導入を基本にしたいものです。

予算の許す限り大容量のメモリ，HDDを選ぼう

選択したマザーボードの仕様で決まるのが，メモリ。現在SDRAM（SDR SDRAM），DDR（DDR SDRAM），RIMMの三種類のメモリ規格があります。最も安価なのが，SDRAMタイプです。しかし，最速でもPC133（バス周波数133MHz）でしかなく，DDRやRIMMなどの最新メモリ規格とは比較にならないほどデータ転送速度が低く，せっかく高性能CPUを導入しても，その足を引っ張ってしまいます。

DDRかRIMMのどちらかの対応のマザーボードとメモリの組み合わせとなるわけですが，このうちRIMMはつい最近までしきりと使用を勧めていたインテル社もDDR対応のチップセットが中心になりはじめ，今後のサポートに不安が残ります。心配であれば，DDR規格のメモリを選んでおけば，高速性能と十分なサポートの両方で満足できるはずです。メモリ容量は，最低256MB，512MBがベストと思ってください。

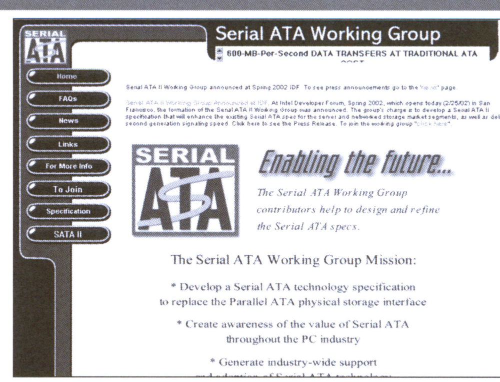
HDDについては，「ウルトラATA/100または133」のIDE接続規格が目安です。容量については，予算が許す限り大容量の製品を選びたいものです。最近は1万円足らずで40GBタイプが販売されていますので，このクラスが最低ラインと考えてください。なお，7200回転／秒の高速モデルもありますが，DTVなどの大容量データを高速で扱う用途以外は，あえて選ぶ理由ないと思います。

●最新情報／HDD編

数多いPCパーツのなかでも，急激に低価格化がすすんでいるのが，HDD。ウルトラATA/100や133など，最新IDE接続の採用で，大容量データの保存や呼び出しにもストレスを感じることなく高速で行えるようになっています。また，保存容量も，ビッグドライブと呼ばれる100GB以上の大容量タイプ，7200回転／秒の高速HDDなど多種多様です。最新マザーボードでも，それ以前の旧規格HDDが接続可能ですが，メモリと同様に全体の処理速度低下の要因ともなりうることから，新規にDOS／V機を自作する場合は，ぜひ最新モデルのHDDを選ぶべきです。容量は40GB前後のものが，お買い得です。



▲全面にラバーを貼った低雑音タイプのHHDもある



▲次期ATA規格である「シリアルATA」は簡単接続で超高速なデータ転送が可能な注目株!!

3 用途に合わせたマシン設計を行おう

完成したものの，実際に使っていたところ，機能や性能に物足りなさを感じてしまうことも珍しくありません。事前に目的とする仕様を明確に出しておかなかったのが，失敗の要因になることも多いようです。また予算についても考慮したいものです。「しまった…」ということがないように，DOS／Vパソコンの主な用途をあらかじめ確認しておきましょう。

主な用途として

①インターネットへの接続
②ワープロや表計算などの学習・ビジネス・ソフト用
③３Ｄゲームを楽しみたい
④ビデオ編集などのAV専用マシンとして使用

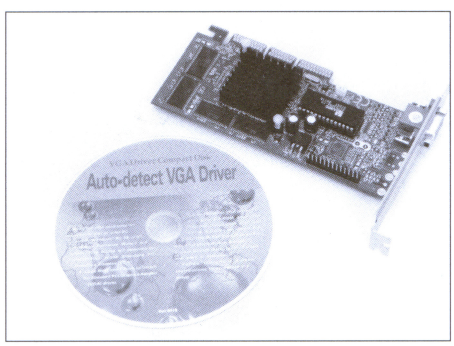
などが挙げられます。①と②については，それほど高性能なCPUは要求されません。CPUに1 GHz前後のペンティアムⅢ，セレロンのほかデュロンを中心とした構成で十分に満足できるはずです。

③は，ベンチマーク・フリークが，最速のマシンを目指すのと同様に，高速であればあるほどベストで，それに見合ったハイエンドＣＰＵや高性能３Ｄグラフィック向けビデオカードなどが求められます。また，ゲームに臨場感を加える高音質サウンドカードも欠かせません。こうなると，必然的に2 GHz近くのペンティアム４やアスロンXPとRIMMまたはDDRメモリの組み合わせを選択することになり，予算も①や②の数倍程度を見ておきたいものです。

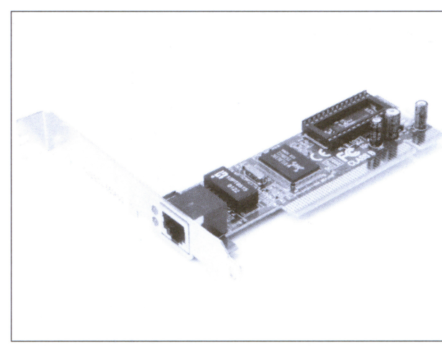
④は，ビデオデッキやデジタル・ビデオカメラとの連携を考え，ビデオキャプチャ・ボードなどのインターフェースの拡充が必要です。また，高度なビデオ編集をスムーズに行える，高速なCPUも欲しくなります。このように，DOS／Vマシンの自作といっても，さまざまな選択肢があるわけで，それに掛かる費用もさまざまです。ですから，パーツ購入前に完成後の用途を明確にしておかなければならないわけです。このあたりの見分け方は，**第１表**を基準にしていただければ良いでしょう。次回は，目的別のパーツ選びと製作のコツを紹介します。

《第１表》用途別DOS／V機の選択基準

用途	パーツ選びのポイント	大体の費用
インターネット接続向け（ブロードバンド回線）	それほどの高速性能は要求されないが，LAN拡張カードは必須。1GHz前後のCPUで十分	６～７万円が目安
ワープロや表計算などの学習，ビジネス・ソフト向け	こちらも1GHz前後のペンティアムⅢ，セレロン，デュロン・クラスで十分	６～７万円が目安
3Dゲーム中心	高クロックのペンティアム４やアスロンXPとRIMM，DDRメモリの組み合わせ	20万円近くと高価で3Dビデオカードも必要
DTVなどのAV専用機向け	ビデオキャプチャ・カードや専用インターフェースを追加	15万円程度は必要

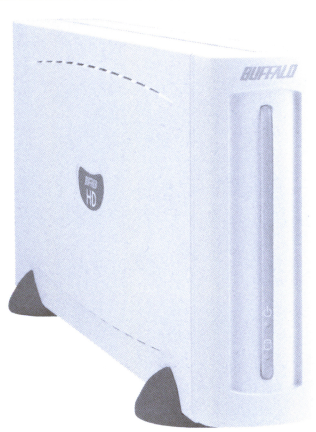


▲3Dゲーム中心の用途であれば，高速な3D描画が可能なビデオカードが不可欠

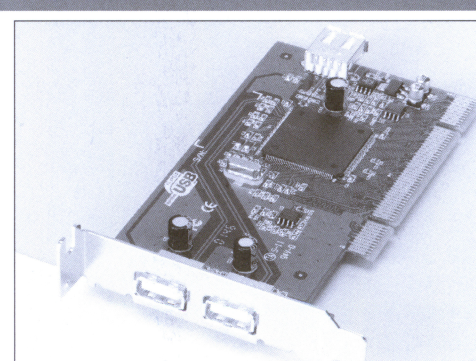

▲ADSLやCATVのブロードバンド接続には，LANカードの増設が必要

●最新情報／インターフェース編

HDDやCD-Rドライブなどの外付け周辺機器の利用には，インターフェースの拡充が不可欠。さまざまなインターフェース規格がありますが，なかでも注目されているのが，「USB2.0」規格。従来のUSB1.1規格を大幅に拡張し，最大450Mbps（メガビット／秒）ものデータ転送規格を持つ超高速インターフェースです。最近では，この規格を採用した周辺機器が続々登場し，頼もしい限りですが，マザーボードでの対応は少数派で，拡張ボードでの対応となります。このように，ボードの増設で，最新インターフェースが簡単に導入できるのも，自作DOS／V機の特徴です。



▶USB2・0の周辺機器が続々登場しているが，自作パソコンでも導入可能だ

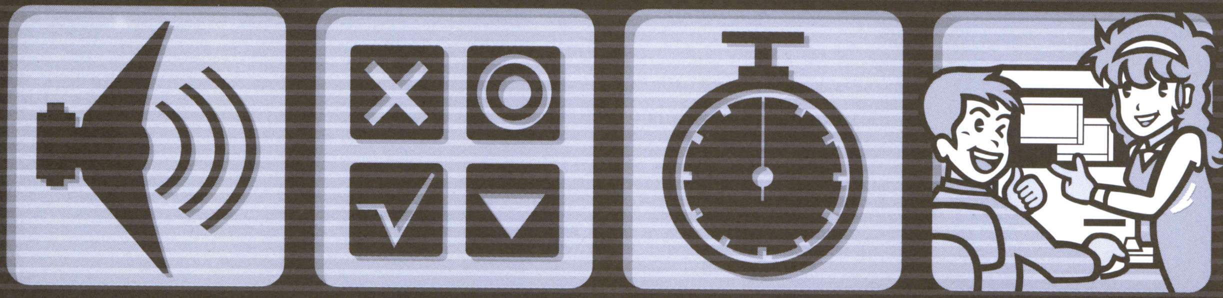

▲起動中でもケーブルの抜き差しが行え，高速なデータ読み書きが可能なUSB2.0拡張カード



本コーナーは，さまざまな言語に対応したプログラミング学習と創作をテーマに，読者からの投稿作品で成り立っています。これから，プログラミングを勉強しようというみなさんや，プロのプログラマー／クリエイターを目指そうというみなさんにとって役立つ情報が満載です。手近にあるパソコンに入力して，お楽しみください。

それぞれの記事にはプログラムを入力／実行するときの注意点やプログラムの内容，改造方法などが紹介されていますので，よく読んでください。また，“CHECKER FLAG”というコラムがあり，デバッグ博士ことDr.Dがそれぞれのプログラムに対し批評を加えています。口が悪い人なので，かなりキビシイことも言っていますが，自分でプログラムを作ったり，これから投稿しようとしている人にとっては，とても参考になるでしょう。

BM特選 プログラマー&クリエイターへの登竜門 投稿プログラムコーナー



プログラム投稿者へのお願い

自分の作ったプログラムを投稿しようという人はつぎのきまりを守って応募してください。

①プログラム（作品）ファイル ②プログラムの説明，住所・氏名・年齢・職業（学校名）・電話番号（連絡のつく時間も）を書いた原稿のテキスト・ファイルをフロッピーディスクなどにセーブし（プリンタで打ち出した原稿も同封のこと），お送りください。

プログラムの説明には，①内容 ②使いかた・遊びかた ③操作方法 ④プログラムのしくみ（変数表やフローチャートも）⑤プログラムの具体的入力方法，注意点 ⑥改造のアドバイス，作ったとき苦労した点 ⑦使用機種名 ⑧言語（バージョン，モードなども）などを書きます。

注1）プログラムは必ずチェックし，ミスを防ぐために同じものを2〜3入れてもらえば安心です。

注2）プログラムの説明は，テキスト・ファイルをフロッピーディスクなどにセーブし，念のためプリンタで打ち出したものも同封することが基本ですが，手書き原稿でも受け付けます。その際，できるだけていねいに，マス目のある用紙を使用してください。

注3）ディスクのラベル，封筒の裏面にもプログラム名（ファイル名も），原稿ファイル名，住所，氏名，電話番号を必ず記入してください。

内容はオリジナルのもので，未発表のものに限ります。もし，参考にした記事や本がありましたら，書名，号数，記事名，筆者名，記事のタイトルなども明記しておいてください。

お送りいただいたディスクはお返しできませんので，あらかじめバックアップをとっておいてください。

掲載されたプログラムに対しては，規定の原稿料をお支払いし，掲載誌をお送りいたします。

編集部

★本誌とじ込みの愛読者カードについている応募カードも忘れずに同封してください。

あて先：〒141-8715東京都品川区東五反田1-11-15 電波新聞社マイコンBASICマガジン「特選プログラム・コーナー」係

掲載プログラムのダウンロード／投稿はこちら

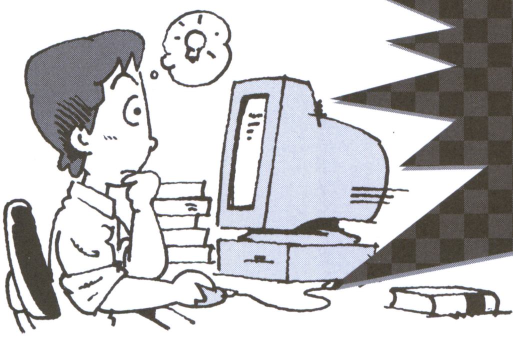
http://www.basicmagazine.net/

Windows(Visual Basic V6.0)用

テキストファイルでアクションゲーム制作

Action

アクション



サバ柄（13歳）

内容

アクションゲームです。と言っても，ただ遊べるというだけでなく，テキストファイルを書き換えることで，ゲーム内容の変更が楽に行えるのが特徴です。

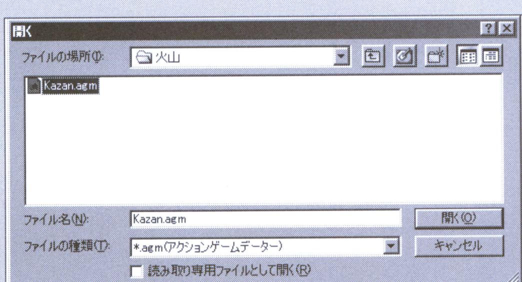
使い方・遊び方

起動すると，ファイルを開くダイアログ画面が出てきます。やりたいゲームファイル(*.agmファイル。ファイル内容は後述)を選択してください(**写真1**)。なお，ゲーム中にファイルの読み込みは出来ません。

ファイルを読み込んだあと，[F5]キーを押すとスタートします(**写真2**)。キャラクターを[←][→]キーで動かしてください。スペースキーを押すとジャンプします(**写真3～4**)。

なお，メニューのショートカットキー設定は，以下のようになっています。

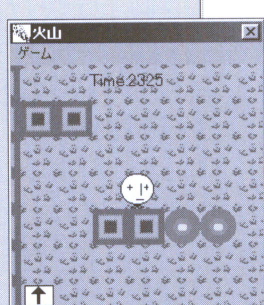
[F5]キー：ゲームスタート，[F6]キー：ゲーム読み込み，[F2]キー：ウェイト調整です。



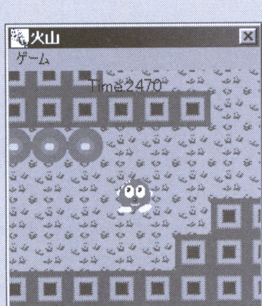
《写真1》最初に，遊びたいアクションゲームデータを選択



《写真2》選択したゲームが読み込まれ，実行される



《写真3》こんな感じのアクションゲームが遊べる



《写真4》今月掲載の『PULL NUMBER』のキャラクターに変更してみた

プログラムの入力方法

プロパティ表を参考にフォームを設定し，掲載されている**フォーム**と**モジュールのリスト**を入力してください。サンプルゲームを遊びたい場合は，掲載されているゲームデータファイルも入力してください。

ゲーム用データの作り方

はじめに，アクションゲームファイル(拡張子が*.agm，**リスト3**もあわせて参考にしてください)を作成します。

メモ帳などのテキストエディタで，以下の形式に沿って，データを書いてください。

"ゲームタイトル",タイトルX座高,"仮想画面ファイル","画像ファイル",タイム,マップ縦,マップ横

・**ゲームタイトル**：全角で6文字まで。半角で12文字まで

・**タイトルX座高**：タイトルの横位置を設定します。適当に決めてください。

・**仮想画面ファイル**：仮想画面が書い

《OVER FLOW》みなさん，ゴールデンウイークはいかがでしたか？新しい環境のもと，疲れた体と心をリフレッシュできましたか？まだまだという人は，本誌を読んでじっくりリフレッシュしてください。2002年後半にかけて新しい企画も用意しています。お楽しみに！

CHECKER FLAG

Dr.D：これは，なかなかユニークな作品じゃ。ゲームそのものでなく，いわばデータ化されたゲームをプレイするためのソフトと言える。ゲームに必要なデータをテキストで作成する形式なので，ゲームそのものの内容は，大きくは変化しないのじゃが，やはり手軽にオリジナルゲームが作れるプログラムという面では評価したい作品じゃな。
つぐ美：オリジナルゲームを制作したいけど，プログラムは苦手って人は，このソフトからゲーム制作をはじめてみてもいいかもしれないわね。
編：より細かな設定ができるようにした，アクションデータ作成用のソフトを作っても面白いかもしれませんね。

てあるファイルを，拡張子付きで書いてください(.amp)。

・**画像ファイル**： 画像ファイルを拡張子付きで書いてください(.bmp)。

・**タイム**：適当に半角数字で入力してください。普通は2000～3000です。

・**マップ縦**：マップの縦の数を入力してください。

・**マップ横**：マップの横の数を入力してください。

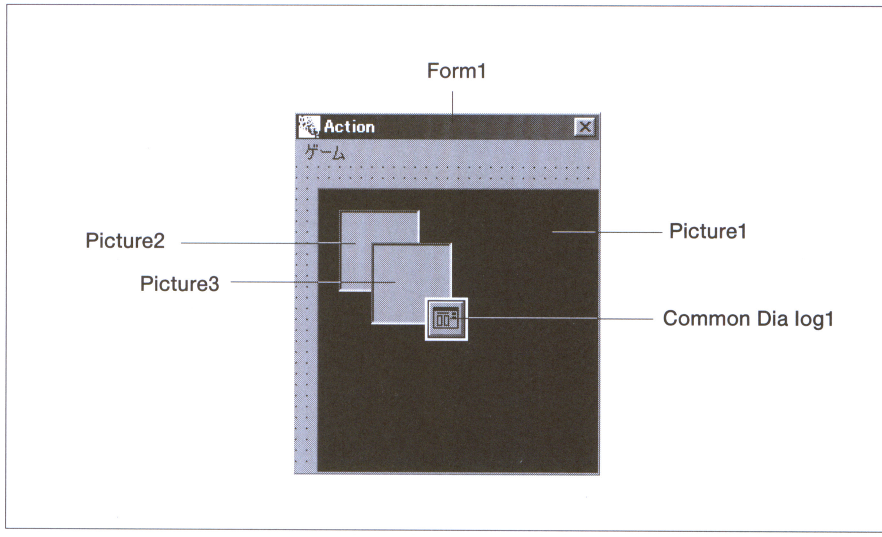
次に，仮想画面データファイル(拡張子が.amp，**リスト4**も合わせて参考にしてください)の作成です。

画面サイズは，60×60が最大です。それ以上の数値を入力するとオーバーフローします。マップデータ内容は，以下のようになります。

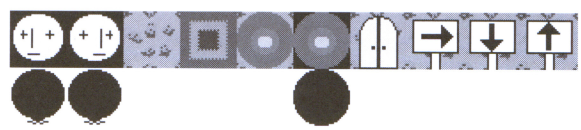
0：何もない空間，1：自分の初期位置，2：壁，床，天井，3：固定の敵，4：空中をバウンドする敵，5：ゴール

4の,空中をバウンドする敵(動く敵)は，最大で3個まで使えます。

最後に，使用する画像ファイルの作り方についてです(kazan.gifもあわせて参考にしてください)。ペイントなどの画像作成ができるソフトで，32×32のサイズの画像を作成し，左から，自左，自右，空間，壁(床，天井)，敵，ゴールとなります。保存はGifでもOKです。



《第1図》フォームの配置図



《第2図》kazan.gifの画像

作者からのコメント

アクションゲームデータの参考になると思って，前回の投稿作品『火山』のアクションゲームバージョンを作りました。たくさんのアクションゲームを作って遊んでください。『火山』を元に，敵を増やしたり，効果音を入れてもいいと思います。

最後に

このソフトを作っている途中に「たくさんのゲームを読み込めるようにしたら面白いかもしれない」と思って，読み込めるようにしました。

あと，これからもアクションゲームのデータを作りたいと思うので，http://www.aa.aeonnet.ne.jp/~aubrey/A/に来てください。ゲーム作成支援ツールもあります。

○付属ゲーム『火山』の内容

火山の中にいるキャラクターをゴール（黄色の扉）まで行かせます。ゴールは黄色のドアです。敵は火の玉です。当らないように気をつけましょう。

★参考にしたもの
・本誌2001年5月号「Get&Run」
・本誌2000年11月号「PaZoo」(どくだみ作)

《第1表》メニューの設定

ｷｬﾌﾟｼｮﾝ(P)	名前(M)	ｲﾝﾃﾞｯｸｽ(X)	ﾍﾙﾌﾟｺﾝﾃｷｽﾄ番号(H)	ｼｮｰﾄｶｯﾄ(S)	ﾈｺﾞｼｴｰﾄ位置(O)	ﾁｪｯｸ(C)	有効 (E)	表示(V)	ｳｲﾝﾄﾞｳﾘｽﾄ(W)
ゲーム	M_Game		0	(なし)	0ーなし		○	○	
…スタート	M_Start		0	F5	0ーなし			○	
…読み込み	M_Load		0	F6	0ーなし		○	○	
…ウェイト	M_Weit		0	F2	0ーなし		○	○	
…-	hr		0	(なし)	0ーなし		○	○	
…終了	M_End		0	(なし)	0ーなし		○	○	

★ベーマガのライバル誌って何ですか？（茨城県古河市・**年齢不相応42歳**）……【**影**：ベースボールマガジン。略してベーマガ。**編**：昔からのネタじゃないですか。**影**：他にライバル誌と呼べる物見当たらないだろう。**編**：独自路線を突っ走るベーマガです。よろしく！】

《第2表》プロパティ表

フォームの設定	
Name	Form1
BorderStyle	1 '固定(実線)
Caption	Action
Height	4020
Icon	(ｱｲｺﾝ)
Left	3675
LinkTopic	Form1
MaxButton	0 'False
MinButton	0 'False
ScaleHeight	224
ScaleMode	3 'ﾋﾟｸｾﾙ
ScaleWidth	224
Top	2055
Width	3450
ピクチャボックスの設定	
Name	Picture1
AutoRedraw	-1 'True
BackColor	&H00000000&
Height	3420
Left	240
ScaleHeight	224
ScaleMode	3 'ﾋﾟｸｾﾙ
ScaleWidth	224
TabIndex	0
Top	240
Visible	0 'False
Width	3420
Name	Picture2
AutoRedraw	-1 'True
Height	900
Left	480
LinkTimeout	0
ScaleHeight	56
ScaleMode	3 'ﾋﾟｸｾﾙ
ScaleWidth	56
TabIndex	1
Top	480
Visible	0 'False
Width	900
Name	Picture3
AutoRedraw	-1 'True
AutoSize	-1 'True
Height	900
Left	840
ScaleHeight	56
ScaleMode	3 'ﾋﾟｸｾﾙ
ScaleWidth	56
TabIndex	2
Top	840
Visible	0 'False
Width	900
コモンダイアログの設定	
Name	CommonDialog1
CancelError	-1 'True
Filter	*.agm(アクションゲームデーター)¦*.agm¦*.*(すべてのファイル)¦*.*
Left	1440
Top	1440

《リスト1》フォームのプログラム・リスト

```
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
   KeySw(KeyCode) = True  'キーフラグオン
End Sub

Private Sub Form_KeyUp(KeyCode As Integer, Shift As Integer)
   KeySw(KeyCode) = False  'キーフラグオフ
End Sub

Private Sub Form_Paint()
  Call BitBlt(Form1.hDC, 0, 0, 224, 224, Picture1.hDC, Px - 96, Py - 96, vbSrcCopy) '
フォームに画像転送
End Sub

Private Sub Form_Unload(Cancel As Integer)
  GameEnd = True
  End
End Sub

Private Sub M_End_Click()
  Unload Form1
End Sub

Private Sub M_Load_Click()
  Call ActionOpen
End Sub

Private Sub M_Start_Click()
  Call Init
End Sub

Private Sub M_Weit_Click()
  Call Weit
End Sub
```

★B（ビビル）L（エルボー）B（ボーン）。（千葉県山武郡・**野潤上15歳**）……【**編**：はて？BLBって何でしょう。**つぐ美**：マクド○ルドのメニューにBLBトマトとかなかったっけ？**編**：ウルティマオンラインに関係してるとか？**影**：フフフ，私だけが知っている。】

《リスト2》モジュールのプログラム・リスト

```
'*****************************
'**********<Action>**********
'********by Sabagara*********
'***********作成日***********
'*******2002:1/9:(土)*******

Option Explicit '変数宣言強制

Public Declare Function timeGetTime Lib "winmm.dll" () As Long 'timeGetTimeを宣言
Public Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal X As Long, ByVal
Y
As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal xSrc As
Long, ByVal ySrc As Long, ByVal dwRop As Long) As Long 'BitBltを宣言
'ゲーム共通変数
Public Px As Integer, Py As Integer '自キャラ用XY座標変数
Public Pdx As Integer, Pdy As Integer  '自キャラ用XY移動速度変数
Public Pjp As Boolean 'ジャンプ用フラグ
Public Pa As Integer 'キャラクターの向き
Public ScreenX As Integer, ScreenY As Integer 'スクリーン
Public Esw(2) As Boolean '敵キャラ用実行変数
Public Ex(2) As Integer, Ey(2) As Integer '敵キャラ用XY座標変数
Public Edx(2) As Integer, Edy(2) As Integer '敵キャラ用移動速度変数
Public KeySw(256) As Byte 'キー入力格納変数
Public WtTime, Wt As Integer 'ウェイト
Public GameEnd As Boolean 'ゲーム終了フラグ
 'ゲーム別変数
Public Map() As Integer  '仮想画面用配列変数
Public Gtitle As String, Mfile As String 'タイトルとマップファイル変数
Public Pfile As String '画像ファイル変数
Public Fx As Integer 'タイトルサイズ用変数
Public GHeight As Integer, GWidth As Integer '画面の高さと幅
Public Spx As Integer, Spy As Integer '自キャラ初期位置
Public SGameTime As Integer, GameTime As Integer  '制限時間
'雑用変数
Public A As Integer, B As Integer, GmFilePath As String
Public I As Integer, J As Integer, K As Integer
Public LineMap As String

Public Const MaxPdx As Integer = 4 '移動最高速度設定
Public Const MaxPdy As Integer = 10 '落下最高速度設定

Public Sub Main()
  Form1.Show
  For K = 0 To 2: Edx(K) = 3: Edy(K) = 3: Next K
  Wt = 25: Px = 96: Py = 96
  Call ActionOpen
End Sub

Public Sub ActionOpen()
On Error GoTo Error
  With Form1
      .CommonDialog1.ShowOpen
    .M_Start.Enabled = False
    Open .CommonDialog1.FileName For Input As #1
      Input #1, Gtitle, Fx, Mfile, Pfile, SGameTime, GHeight, GWidth
    Close #1
    .Picture1.Height = GHeight * 32 + 4: .Picture1.Width = GWidth * 32 + 4
    .Picture2.Height = GHeight * 32 + 4: .Picture2.Width = GWidth * 32 + 4
    GHeight = GHeight - 1: GWidth = GWidth - 1
    ReDim Map(GWidth, GHeight) As Integer  '仮想画面用配列変数
    GmFilePath = Left(.CommonDialog1.FileName, (Len(.CommonDialog1.FileName) -
Len(.CommonDialog1.FileTitle)))
    .Caption = Gtitle
    .Picture3.Picture = LoadPicture(GmFilePath & Pfile)
    Open GmFilePath & Mfile For Input As #1
      For A = 0 To GHeight
        Line Input #1, LineMap
        For B = 0 To GWidth
          Map(B, A) = Val(Mid(LineMap, B + 1, 1))
        Next B
      Next A
    Close #1
    For K = 0 To 2: Esw(K) = False: Next K: K = 0
    For A = 0 To GHeight
      For B = 0 To GWidth
        Select Case Map(B, A)
        Case 0: Call PutBack(2)     '空間
        Case 1: Spx = B * 32: Spy = A * 32: Call PutBack(2) '初期位置
        Case 2: Call PutBack(3)    '床
        Case 3: Call PutBack(4)    '敵
        Case 4: Ex(K) = B * 32: Ey(K) = A * 32: Esw(K) = True: K = K + 1: Call PutBack(2)
'動く敵
        Case 5: Call PutBack(6) 'ゴール
        Case Else: Call PutBack(Map(B, A))
        End Select
      Next B
    Next A
    .M_Start.Enabled = True
    Call Title
    Exit Sub
  End With
Error:
  If Err <> 32755 Then
    MsgBox ("読み込み時にエラーが発生しました。")
    Unload Form1
  End If
End Sub

Public Sub PutBack(PutM)
  Dim PutX As Integer, PutY As Integer
  PutX = B * 32: PutY = A * 32
  Call BitBlt(Form1.Picture2.hDC, PutX, PutY, 32, 32, Form1.Picture3.hDC, PutM * 32, 0,
vbSrcCopy)
End Sub

Public Sub GameMain() 'ゲームメインループ
  Do Until GameEnd = True 'ゲームフラグ監視
    DoEvents '他処理監視
    WtTime = timeGetTime 'ウェイトタイム測定
    '**********WithをForm1にあてはめる
    With Form1
    '***自キャラ操作&背景当たり判定***
```

★そういえばベーマガでは「P/ECE」ではなく「PIECE」なんですね。（青森県南津軽郡・**ていこく23歳**）……【**編**：本誌では，ロゴとして扱う場合は，/ですが，読みは「ピース」なので，Iで統一しています。**つぐ美**：アクアプラスさんの了解も取っているのよ。】

```
      If KeySw(39) Then  '右
        Pdx = Pdx + 1: Pa = 1
        If Pdx > MaxPdx Then Pdx = MaxPdx
      ElseIf KeySw(37) Then '左
        Pdx = Pdx - 1: Pa = 0
        If Pdx < -MaxPdx Then Pdx = -MaxPdx
      Else
        Pdx = Pdx - Sgn(Pdx)
      End If
      Px = Px + Pdx
      If Map(Px ¥ 32, Py ¥ 32) = 2 Or Map(Px ¥ 32, (Py + 31) ¥ 32) = 2 Then _
        Px = (Px ¥ 32) * 32 + 32: Pdx = 0
      If Map((Px + 31) ¥ 32, Py ¥ 32) = 2 Or Map((Px + 31) ¥ 32, (Py + 31) ¥ 32) = 2 Then
_
        Px = (Px ¥ 32) * 32: Pdx = 0
      If KeySw(32) = True And Pjp = False Then 'スペース
        Pdy = -15
      End If
      Py = Py + Pdy: If Pdy < MaxPdy Then Pdy = Pdy + 1
      If Map(Px ¥ 32, Py ¥ 32) = 2 Or Map((Px + 31) ¥ 32, Py ¥ 32) = 2 Then _
        Py = (Py ¥ 32) * 32 + 32: Pdy = 0
      If Map(Px ¥ 32, (Py + 31) ¥ 32) = 2 Or Map((Px + 31) ¥ 32, (Py + 31) ¥ 32) = 2 Then
        Py = (Py ¥ 32) * 32: Pdy = 0: Pjp = False
      Else
        Pjp = True
      End If
    '************動く敵**************
      Call EMove
    '**********背景処理**************
      Call BitBlt(.Picture1.hDC, Px - 96 - 32, Py - 96 - 32, 288, 288, .Picture2.hDC, Px
- 96 - 32, Py - 96 - 32, vbSrcCopy)
    '********キャラ画像転送***********
      Call SendChar
    '**********当たり判定************
      For K = 0 To 2
        If Px + 25 > Ex(K) And Px - 25 < Ex(K) And Py + 25 > Ey(K) And Py - 25 < Ey(K)
Then Call GameOver(0): Exit Do
      Next K
      GameTime = GameTime - 1
      If GameTime <= 0 Then Call GameOver(1): Exit Do
      If Map(Px ¥ 32, Py ¥ 32) = 5 Then Call GameClear: Exit Do 'ゴール
      If Map(Px ¥ 32, Py ¥ 32) = 3 Or Map(Px ¥ 32, (Py + 31) ¥ 32) = 3 Or Map((Px + 31) ¥
32, Py ¥ 32) = 3 Or _
      Map((Px + 31) ¥ 32, (Py + 31) ¥ 32) = 3 Then Call GameOver(0): Exit Do  '敵にぶつ
かる
    '**********文字転送**************
      Call PrintInp(Px - 96 + 70, Py - 96 + 5, 12, RGB(0, 0, 0), "Time:" & GameTime)
    '**********画像転送**************
      Call BitBlt(Form1.hDC, 0, 0, 224, 224, .Picture1.hDC, Px - 96, Py - 96, vbSrcCopy)
'フォームに画像転送
    '**********ウェイト**************
      While (WtTime + Wt > timeGetTime): DoEvents: Wend  'ウェイト
    '**********Withを閉じる**********
    End With
  Loop
End Sub
```

```
Public Sub PrintInp(PRX, PRY, PRS, PRCOL, PR) '画面に文字を描く
    With Form1.Picture1
        .FontSize = PRS  'フォントサイズ
        .CurrentX = PRX 'フォントX座標
        .CurrentY = PRY 'フォントY座標
        .ForeColor = PRCOL 'フォントカラー
    End With
    Form1.Picture1.Print PR '文字
End Sub

Public Sub Title() 'タイトル画面
  With Form1
    Px = 96: Py = 96
    .Picture1.BackColor = RGB(255, 255, 255)
    Call PrintInp(82, 152, 18, RGB(176, 176, 176), "Action")
    Call PrintInp(80, 150, 18, RGB(0, 0, 0), "Action")
    Call PrintInp(75, 210, 10, RGB(128, 0, 0), "Programming By Sabagara")
    Call PrintInp(Fx, 70, 32, RGB(0, 0, 0), Gtitle)
    Call BitBlt(Form1.hDC, 0, 0, 224, 224, .Picture1.hDC, 0, 0, vbSrcCopy) 'ﾌｫｰﾑに画像転
送
  End With
End Sub

Public Sub Init()
  Form1.M_Load.Enabled = False
  Px = Spx: Py = Spy: Pa = 1
  Pjp = False: Pdx = 0: Pdy = 0
  Form1.KeyPreview = True
  GameTime = SGameTime
  GameEnd = False
  Call GameMain
End Sub

Public Sub SendChar()
  Call PutChar(Px, Py, Pa * 32, 0, Pa * 32, 32) '自キャラ
  For K = 0 To 2
    If Esw(K) = True Then Call PutChar(Ex(K), Ey(K), 5 * 32, 0, 5 * 32, 32) '敵キャラ
  Next K
End Sub

Public Sub PutChar(X, Y, BX, BY, MX, MY) 'キャラクタを表示する
With Form1
    Call BitBlt(.Picture1.hDC, X, Y, 32, 32, .Picture3.hDC, MX, MY, vbSrcAnd)    'AND合成
    Call BitBlt(.Picture1.hDC, X, Y, 32, 32, .Picture3.hDC, BX, BY, vbSrcInvert) 'XOR合成
End With
End Sub

Public Sub GameClear()
  With Form1
    Call BitBlt(.Picture1.hDC, 0, 0, 960, 1600, .Picture2.hDC, 0, 0, vbSrcCopy)
    Call SendChar
    Call PrintInp(Px - 96 + 50, Py - 96 + 70, 40, RGB(0, 0, 0), "Clear")
    Call PrintInp(Px - 96 + 70, Py - 96 + 5, 12, RGB(0, 0, 0), "Time:" & GameTime)
    Call BitBlt(Form1.hDC, 0, 0, 224, 224, .Picture1.hDC, Px - 96, Py - 96, vbSrcCopy) '
フォームに画像転送
    GameEnd = True
    Form1.M_Load.Enabled = True
```

★プログラミングの勉強目的で，3月号からベーマガ購読を始めた私の決意はまちがってないでしょうか？（神奈川県横浜市・**相模入道19歳**）……【**Dr.D**：もちろん，まちがってなどおらん。プログラミングの登竜門はベーマガだけ。がんばって勉強するんじゃぞ。】

```
  End With
End Sub

Public Sub GameOver(Q)
  With Form1
    Call BitBlt(.Picture1.hDC, 0, 0, 960, 1600, .Picture2.hDC, 0, 0, vbSrcCopy)
    Call SendChar
    If Q = 0 Then
      Call PrintInp(Px - 96 + 50, Py - 96 + 70, 40, RGB(0, 0, 0), "Miss !")
    Else
      Call PrintInp(Px - 96 + 20, Py - 96 + 70, 32, RGB(0, 0, 0), "Time Over")
    End If
    Call PrintInp(Px - 96 + 70, Py - 96 + 5, 12, RGB(0, 0, 0), "Time:" & GameTime)
    Call BitBlt(Form1.hDC, 0, 0, 224, 224, .Picture1.hDC, Px - 96, Py - 96, vbSrcCopy) '
フォームに画像転送
    GameEnd = True
    Form1.M_Load.Enabled = True
  End With
End Sub

Public Sub EMove()
  For K = 0 To 2
    If Esw(K) = True Then
      If Map(Ex(K) ¥ 32, Ey(K) ¥ 32) = 2 Or Map(Ex(K) ¥ 32, (Ey(K) + 31) ¥ 32) = 2 Then _
        Edx(K) = -Edx(K)
      If Map((Ex(K) + 31) ¥ 32, Ey(K) ¥ 32) = 2 Or Map((Ex(K) + 31) ¥ 32, (Ey(K) + 31) ¥
32) = 2 Then _
        Edx(K) = -Edx(K)
      If Map(Ex(K) ¥ 32, Ey(K) ¥ 32) = 2 Or Map((Ex(K) + 31) ¥ 32, Ey(K) ¥ 32) = 2 Then _
        Edy(K) = -Edy(K)
      If Map(Ex(K) ¥ 32, (Ey(K) + 31) ¥ 32) = 2 Or Map((Ex(K) + 31) ¥ 32, (Ey(K) + 31) ¥
32) = 2 Then _
        Edy(K) = -Edy(K)
      If Ex(K) > Form1.Picture1.Width - 68 Or Ex(K) < 32 Then Edx(K) = -Edx(K) '移動範囲
設定
      If Ey(K) > Form1.Picture1.Height - 68 Or Ey(K) < 32 Then Edy(K) = -Edy(K)
      Ex(K) = Ex(K) + Edx(K): Ey(K) = Ey(K) + Edy(K) '敵の移動
    End If
  Next K
End Sub

Public Sub Weit() 'ウェイト設定
On Error Resume Next
    A = InputBox("ウェイトを入力して下さい。0～1000", "ウェイト設定", Wt)
    If Err = 0 Then 'エラーを起こさない時
        If A < 0 Or A > 1000 Then '有効範囲以外
            Exit Sub 'プロシージャをぬける
        End If
        Wt = A '値を設定
    End If
End Sub
```

《リスト3》Kazan.agmのリスト

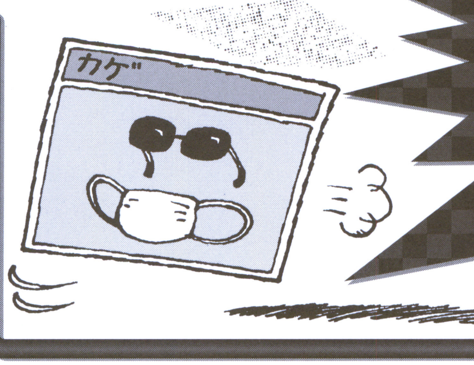
```
"火山",70,"Kazan.amp","Kazan.gif",3000,50,40
```

《リスト4》Kazan.ampのリスト

```
222222222222222222222222222222
222222222222222222222222222222
222222222222222222222222222222
222000000022204000022200000222
222000000002200000022000000222
222000000000200000020000000222
222500000000000000000000000222
222220022222222222222220000222
222000000000000000000000000222
222000000000000000000002200222
222000000000000000000000000222
222000000000000000000000000222
222000000000000000000000022222
222002200220022002200008000222
222000000000002000000222000222
222000000000002000000200000222
222220000000002000000200000222
222000000000002000000200000222
222000000000002000000000000222
222002233003300200000000000222
222000000000000222200000700222
222900000000000000022200200222
222220000000000000000300200222
222220000000000000000300200222
222222222220000000000300200222
222222222222000000000300200222
222222222222222000000300200222
222000000000000000000000200222
222000000000000009000000200222
222000000000000222222200200222
222000000000000000000200200222
222000000000000000000200200222
222000000002222000000200000222
222000000000000000000200000222
222000000000000000000200000222
222000222200000000000200000222
222000000000000000000200000222
222309000000000000332000000222
222222000000000000022200000222
222222200000000000022200000222
222222222222220000022200000222
222222222222222200022200000222
222333333333330000022200000222
222000000000000000022233333222
222100000000000022222233333222
222000700000000022222223333322
222222222222222222222233333222
222222222222222222222233333222
222222222222222222222233333222
```

★iアプリ以外の携帯電話向けプログラム（Javaアプリ，ezplus）の投稿は受け付けているのでしょうか？（秋田市・**福田　拓23歳**）
……【**編**：もちろん受け付けています。どしどし投稿してください。掲載されていないから，これはダメってことはありません。】

Windows(Visual Basic V6.0)用



フォームアクションゲーム

ムーブ

Move

エレフ（17歳）

内容

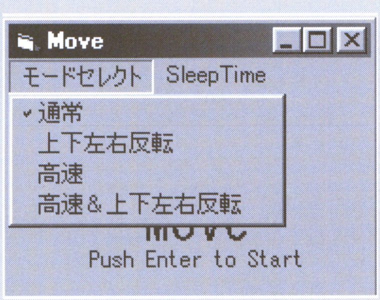
画面上を常に移動しつづけるフォームを,うまくコントロールして，画面外枠に触れないようにするゲームです。

遊び方

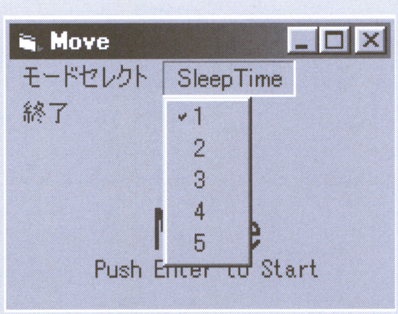
まずモードを選択し,スリープタイムを設定して，Enter キーでスタートします(**写真１，２**)。

カーソルキーを押して，動きつづけるフォームをうまくコントロールして，画面外枠に触れないようにします(**写真３，４**)。

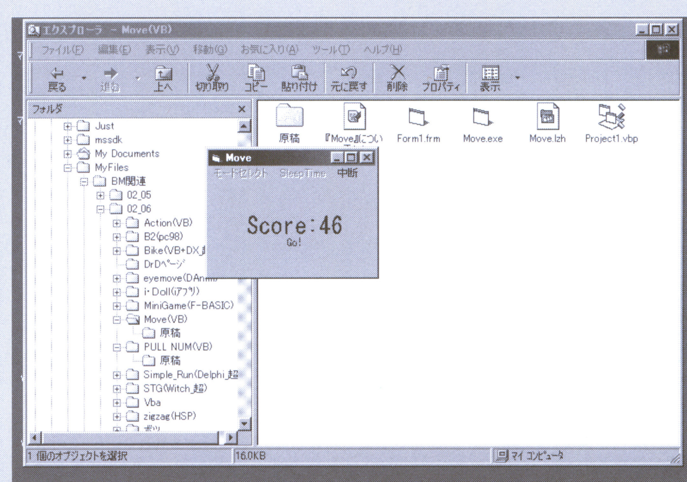
移動には加速度があり，キーを押さないと，その進行方向に移動量が加算され



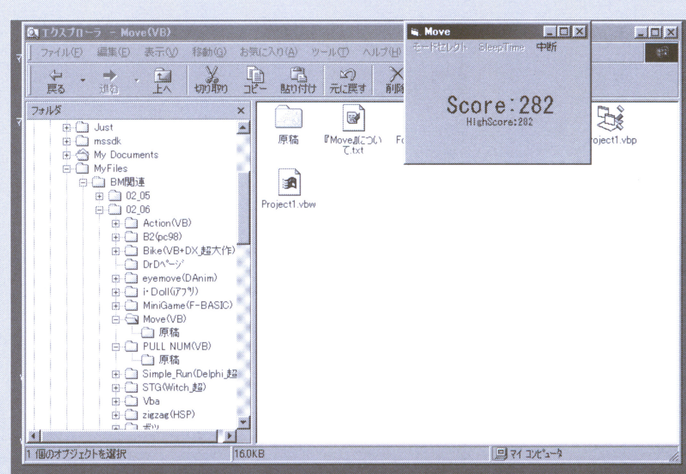
《写真１》速度やキータイプの違いがある四つのモードが選択できる



《写真２》ゲーム全体の速度を決めるSleepTimeも設定できる



《写真３》画面上を移動していくフォームを操作していく



《写真４》画面外へ移動するとダメ。加速度を考慮して操作しよう

★POV-Ray講座っておもしろそうじゃありません？テキスト形式だし投稿もできそう。(熊本県玉名郡・**巨石人20歳**）……【**編**：レイトレーシング法で3DCGを描き出すためのフリーソフトですね。テキストでオブジェクトの形状，配置や色などを指定していくんです。】

ますので，ほっておくとダメです。フォームのサイズが，だんだん大きくなります。

各モードについて

・通常：通常モードです。

・上下左右反転：キー入力の移動方向が逆になります。

・高速：移動量の変化が大きくなります。

・高速＆上下左右反転：キー入力の移動方向が逆になるうえに，移動量の変化が大きくなります。

高速＆上下左右反転モードが一番難しいです。

改造のヒント

・コメントを参考に数値をいじってみましょう。難しくするしかできないと思いますが。

・操作性を改善してみる。

工夫した点

・プレイヤーが翻弄されるような動きをさせるところで手間どりました。

・ゲームオーバー時に1秒停止させて，フォームが枠にぶつかったことをわかりやすくしました。

CHECKER FLAG

編：最近では少なくなってきましたが，こういったVBのフォーム自体を使った作品というのは，本誌でVBプログラムを掲載し始めた頃に，けっこうありましたね。

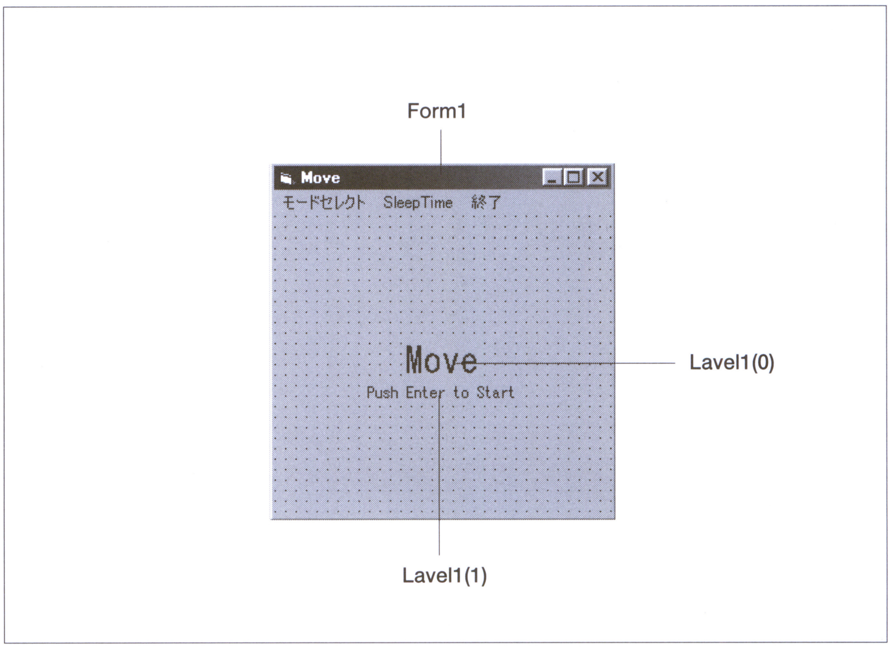
影：内容は単純だけど，加速度のついた動きがしっかり作られている感じだ。

Dr.D：リスト面も，スリープタイムの設定やウィンドウの大きさ設定の部分で，チョットした工夫がされておるが，他は単純な形式で作られているので，プログラム初心者には参考にしやすいリストと言えるじゃろう。

つぐ美：プレイ時間も短く，手軽に遊べるゲームと言えるわね。

《第１表》プロパティ表

フォームの設定	
Name	Form1
Caption	Move
Height	4110
Left	105
LinkTopic	Form1
ScaleHeight	3420
ScaleWidth	3855
StartUpPosition	3 'Windows の既定値
Top	105
Width	3975
ラベルの設定	
Name	Label1(0)
Alignment	2 '中央揃え
Caption	Move
Font	ＭＳ ゴシック
…Size	20
Height	405
Index	0
Left	1500
TabIndex	0
Top	1440
Width	855
Name	Label1(1)
Alignment	2 '中央揃え
Caption	Push Enter to Start
Font	ＭＳ ゴシック
…Size	9
Height	180
Index	1
Left	1065
TabIndex	1
Top	1920
Width	1725



《第1図》フォームの配置図

《第２表》メニューの設定

ｷｬﾌﾟｼｮﾝ(P)	名前(M)	ｲﾝﾃﾞｯｸｽ(X)	ﾍﾙﾌﾟｺﾝﾃｷｽﾄ番号(H)	ｼｮｰﾄｶｯﾄ(S)	ﾈｺﾞｼｴｰﾄ位置(O)	ﾁｪｯｸ(C)	有効（E)	表示(V)	ｳｲﾝﾄﾞｳﾘｽﾄ(W)
モードセレクト	ModeSelection		0	（なし）	0－なし		○	○	
…通常	Modes	0	0	（なし）	0－なし	○	○	○	
…上下左右反転	Modes	1	0	（なし）	0－なし		○	○	
…高速	Modes	2	0	（なし）	0－なし		○	○	
…高速＆上下左右反転	Modes	3	0	（なし）	0－なし		○	○	
SleepTime	Sleeps		0	（なし）	0－なし		○	○	
…1	Sleep	0	0	（なし）	0－なし	○	○	○	
…2	Sleep	1	0	（なし）	0－なし		○	○	
…3	Sleep	2	0	（なし）	0－なし		○	○	
…4	Sleep	3	0	（なし）	0－なし		○	○	
…5	Sleep	4	0	（なし）	0－なし		○	○	

★ノートレポート：Meが不安定というよりはブラウザが不安定という結論に。現在，脱IE化を画策中。（岩手県釜石市・**鉄人坂井24歳**）……【**編**：あれ？なんかこのパソコン不安定だね。**編集長**：さっきまで影君が使っていたからな。**影**：私はウイルスか！】

《リスト》Moveのプログラム・リスト

```
Option Explicit

Dim X As Long, Y As Long 'フォームの位置
Dim Xs As Long, Ys As Long 'フォームの移動量
Dim Score As Long, Highscore As Long 'スコア,ハイスコア
Dim ExitGameFlag As Boolean '中断フラグ
Dim SleepTime As Integer 'スリープタイム
Dim Mode As Integer 'ゲームモード
Dim Plus As Integer '移動量の変化量

'Windows API
Private Declare Function GetTickCount& Lib "Kernel32" ()

Dim Wtime As Long 'GetTickCount用

Private Declare Function GetKeyState Lib "user32" (ByVal nVirtKey As Long) As Long

Private Sub ExitGame_Click()

    If ExitGame.Caption = "中断" Then
        ExitGameFlag = True
    Else
        End
    End If

End Sub

Private Sub Form_Load()

    '初期設定
    Randomize
    Me.Height = Screen.Height / 4
    Me.Width = Screen.Width / 4
    Me.Left = Int(Screen.Width - Me.Width) / 2
    Me.Top = Int(Screen.Height - Me.Height) / 2
    SleepTime = 0
    Plus = 2
    Me.Show

    Do
        Wtime = GetTickCount

        If (GetKeyState(vbKeyReturn) And &H80) <> 0 Then
            'ゲーム開始時の初期設定
            Score = 0
            X = Me.Left
            Y = Me.Top
            Xs = -1 ^ (Int(Rnd * 4) + 1)
            Ys = -1 ^ (Int(Rnd * 4) + 1)
            ExitGame.Caption = "中断"
```

```
            ExitGameFlag = False
            ModeSelection.Enabled = False
            Sleeps.Enabled = False
            While Wtime + 1000 > GetTickCount
                DoEvents
                Label1(1).Caption = "Ready..."
            Wend
            Label1(1).Caption = "Go!"
            Do
                Wtime = GetTickCount

                'キー入力判定
                If (GetKeyState(vbKeyRight) And &H80) <> 0 Then
                    Xs = Xs + Plus
                End If
                If (GetKeyState(vbKeyLeft) And &H80) <> 0 Then
                    Xs = Xs - Plus
                End If
                If (GetKeyState(vbKeyDown) And &H80) <> 0 Then
                    Ys = Ys + Plus
                End If
                If (GetKeyState(vbKeyUp) And &H80) <> 0 Then
                    Ys = Ys - Plus
                End If

                Call subPlus
                If Mode >= 2 Then Call subPlus

                '最終的なフォームの位置取得
                X = X + Xs
                Y = Y + Ys
                Me.Left = X
                Me.Top = Y
                Me.Height = Me.Height + 1
                Me.Width = Me.Width + 1

                'スコア処理
                Score = Score + 1
                Label1(0).Caption = "Score:" & Score
                If Score > Highscore Then Highscore = Score
                If Score > 50 Then
                    Label1(1).Caption = "HighScore:" & Highscore
                End If

                'ゲームオーバー判定
                If subFormCheck = True Or ExitGameFlag = True Then Exit Do

                Call subSleep
            Loop
            Wtime = GetTickCount
            While Wtime + 1000 > GetTickCount
                DoEvents
```


★春休みに免許取得！中古の車を買いました。大学に車で通学の毎日です。影さんも隣に乗せてあげようか？ドアがはずれそうだけど。（福島県郡山市・ゾーン20歳）……【影：毎日ベンツ通勤の私としては，遠慮しとくわ。編：影さんの場合，山手線もベンツ？！】

```
            Wend
            Me.Height = Screen.Height / 4
            Me.Width = Screen.Width / 4
            Me.Left = Int(Screen.Width - Me.Width) / 2
            Me.Top = Int(Screen.Height - Me.Height) / 2
            ModeSelection.Enabled = True
            Sleeps.Enabled = True
            ExitGame.Caption = "終了"
        End If

        DoEvents
    Loop

End Sub

Sub subPlus()

    'オート移動処理
    If Xs > 0 Then
        Xs = Xs + 1
    ElseIf Xs < 0 Then
        Xs = Xs - 1
    End If
    If Ys > 0 Then
        Ys = Ys + 1
    ElseIf Ys < 0 Then
        Ys = Ys - 1
    End If

End Sub

Private Sub Form_Resize()

    Dim lblFor As Integer

    'ラベルの位置変更
    For lblFor = 0 To 1
        Label1(lblFor).Width = Me.Width
        Label1(lblFor).Left = 0
    Next lblFor
    Label1(0).Top = Int(Me.Height - Label1(0).Height) / 2 - Label1(0).Height
    Label1(1).Top = Label1(0).Top + Label1(0).Height

End Sub

Function subFormCheck() As Boolean

    'ゲームオーバー判定
    If Me.Left < 0 Or Me.Left + Me.Width > Screen.Width Or _
       Me.Top < 0 Or Me.Top + Me.Height > Screen.Height Then
        subFormCheck = True
    End If

End Function

Private Sub subSleep()

    'ウェイト(簡易スリープ)
    If SleepTime = 0 Then DoEvents: Exit Sub
    While Wtime + SleepTime * 10 > GetTickCount
        DoEvents
    Wend

End Sub

Private Sub Form_Unload(Cancel As Integer)

    End

End Sub

Private Sub Modes_Click(Index As Integer)

    Dim f As Integer

    For f = 0 To 3
        Modes(f).Checked = False
    Next f
    Modes(Index).Checked = True
    Mode = Index
    Select Case Index
    Case 0
        Plus = 2
    Case 1
        Plus = -2
    Case 2
        Plus = 4
    Case 3
        Plus = -4
    End Select

End Sub

Private Sub Sleep_Click(Index As Integer)

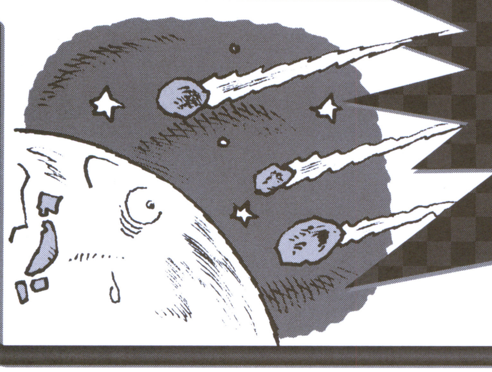
    Dim f As Integer

    For f = 0 To 4
        Sleep(f).Checked = False
    Next f
    Sleep(Index).Checked = True
    SleepTime = Index

End Sub
```

★ま（豆）ご（ゴマ）わ（ワカメ）や（野菜）さ（魚）しい（椎茸）が身体にいいらしいですよ。（福岡市・**前田宣弘28歳**）……**【編：**語呂と言えば周期表の覚えかた。“水兵リーベぼくの船，ソー曲がるシップスクラークか…”。地域などで微妙に違ってるようだけどね。**】**

Windows(Visual Basic V6.0)用



隕石を撃ち落として地球を守れ

METEO（メテオ）

峠　恒司（47歳）

ルール

シューティングゲームです。隕石から地球を守ります。

宇宙から降ってくる隕石をミサイルで撃ち落とします(**写真1，2**)。ミサイルで隕石を撃ち落とすと消えてなくなります。隕石は三つの大きさがあり，いろいろな動きで地球に落ちてきます(**写真3**)。隕石が地球に当たるとゲームオーバーです(**写真4**)。

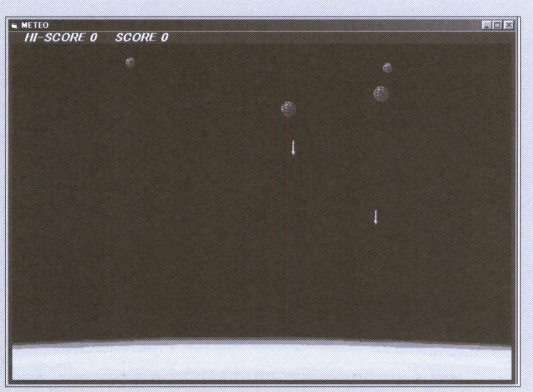
ミサイルで損石を撃った所が地球から離れていればいるほど高得点です。20,000点以上のスコアになるとスペースシャトルが見れます。

操作方法

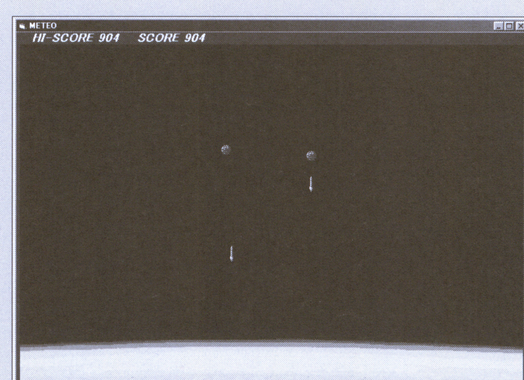
マウスをクリックした所に向かってミサイルが発射されます。ミサイルは，同時に3発しか発射できません。

プログラムについて

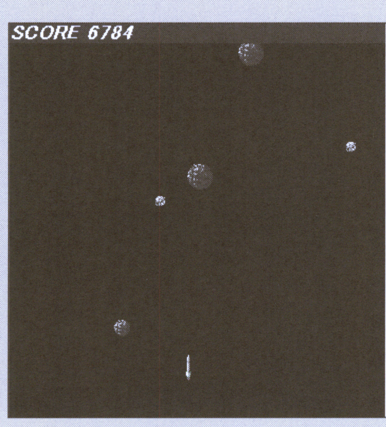
フォームにピクチャーボックスを二つ貼り付けます。一つは表示用，もう一つは隕石やミサイルのキャラクターを納めます。地球のCGは，画面に直接読み込んで描画します。



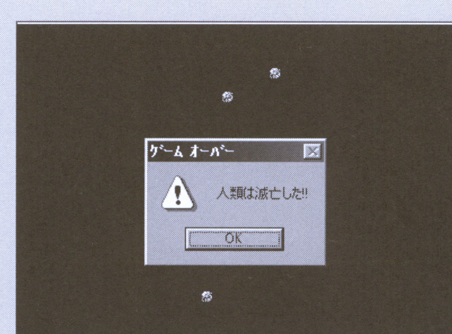
《写真1》接近してくる隕石を撃ち落とせ



《写真2》隕石の動きに狙いを定めろ

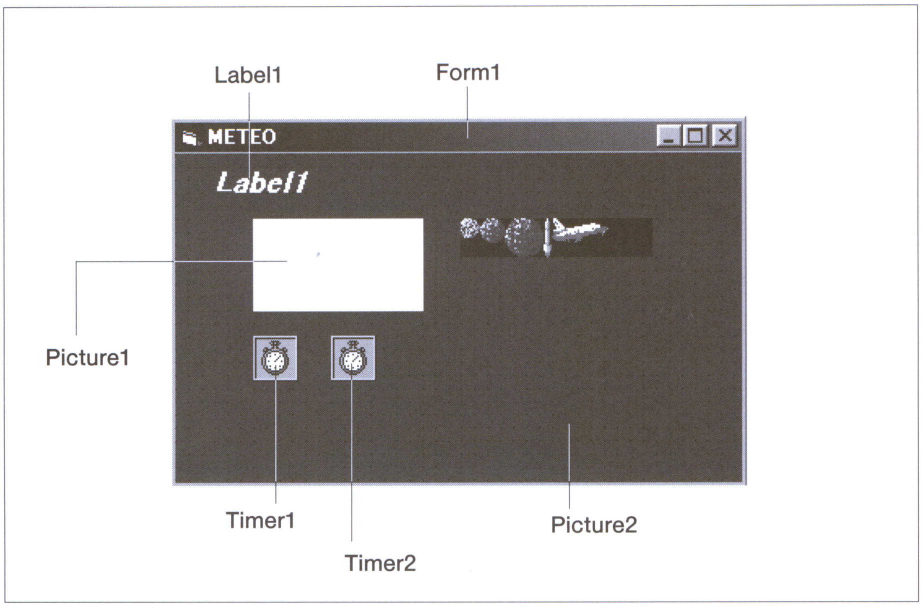


《写真3》小さい隕石を撃ち落とすのが難しい



《写真4》20,000点を目指してプレイ

★4月号から購読していますが，VB初心者の僕には，投稿プログラム・コーナー大変参考になります。こんな雑誌があったとは……。（神奈川県秦野市・**酉龍20歳**）……【**編**：あったんですねー。読者が寄せてくる投稿プログラムは，参考にも，刺激にもなります。】



《第１図》フォームの配置図



《第2図》EARTH. GIFの画像

CHECKER FLAG

つぐ美：遊びやすいシューティングですが，同時に発射できる弾が３発まで，というのが気になりました。やっぱりシューティングは，ある程度弾が連射できないと爽快感がないですね。

影：隕石の動き方は，パターンが多くて工夫していると感じた。やや不自然な動きもあるけど，隕石の移動先に狙いを定めるという要素が生まれているゾ。

Dr.D：つぐ美クンが気になったというミサイル数は，ミサイルの座標を扱う配列の数を増やせば実現できるのじゃが，この配列を扱っているループ文の値も合わせて書き換える必要があるため，記述面でチト面倒じゃな。ミサイルの最大発射数を定数で定義しておいて，その値を書き換えればOKという形で記述してほしかったな。

編：リストは処理を実現させるだけでなく，修正・改良面も考慮して記述してほしいですね。

隕石やミサイルなどは，118×24ドットの大きさのビットマップファイルの中に一緒に描きます。

地球は780×68ドットで描き，GIF形式で保存します(EARTH.GIF)。

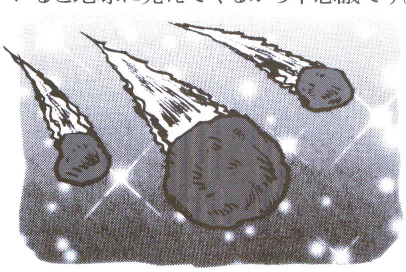
タイマーは二つで，隕石とミサイルを動かします。

WAVファイルはミサイルの発射，爆破，地球に当たった時の音です。イメージに合わせて適当に作ってください。

おわりに

隕石の動き方は17とおりあります。何回かやっているとわかります。２万点めざして頑張ってください。

画面下の地球は,じっと見ていてもそれらしく見えませんが，ゲームをやっていると地球に見えてくるから不思議です。



《第１表》プロパティ表

フォームの設定	
Name	Form1
BackColor	&H00404040&
Caption	METEO
ForeColor	&H00000000&
HasDC	0 'False
Height	3435
Left	0
LinkTopic	Form1
ScaleHeight	202
ScaleMode	3 'ﾋﾟｸｾﾙ
ScaleWidth	349
StartUpPosition	3 'Windows の既定値
Top	0
Width	5355
ピクチャボックスの設定	
Name	Picture1
Appearance	0 'ﾌﾗｯﾄ
AutoRedraw	-1 'True
BackColor	&H00FFFFFF&
BorderStyle	0 'なし
ForeColor	&H80000008&
HasDC	0 'False
Height	57
Left	48
ScaleHeight	57
ScaleMode	3 'ﾋﾟｸｾﾙ
ScaleWidth	105
TabIndex	2
Top	40
Width	105
Name	Picture2
AutoRedraw	-1 'True
AutoSize	-1 'True
BackColor	&H00C0C0C0&
BorderStyle	0 'なし
Height	24
Left	176
Picture	(ﾋﾞｯﾄﾏｯﾌﾟ)
ScaleHeight	24
ScaleMode	3 'ﾋﾟｸｾﾙ
ScaleWidth	118
TabIndex	0
Top	40
Visible	0 'False
Width	118
ラベルの設定	
Name	Label1
Appearance	0 'ﾌﾗｯﾄ
AutoSize	-1 'True
BackColor	&H00404040&
BackStyle	0 '透明
Caption	Label1
Font	ＭＳ Ｐゴシック
…Size	14
…ｽﾀｲﾙ	太字 斜体
ForeColor	&H00FFFFFF&
Height	19
Left	24
TabIndex	1
Top	8
Width	75
タイマーの設定	
Name	Timer1
Left	48
Top	112
Name	Timer2
Left	96
Top	112

★2年からキャンパスが変わるため引っ越しました。で，生協に行ってみるとベーマガがっ！さらには5インチFDまでっ！びっくりです。（群馬県桐生市・トランス19歳）……【影：ベーマガ生協乗っ取り作戦を敢行する。学生読者諸君は，一斉に生協に走れ！】

《リスト１》フォームのプログラム・リスト

```
Option Explicit
Private Sub NEWGAME()

Dim J As Integer, Z As Integer

For M = 1 To 261: For N = 0 To 155: A(M, N) = 0: Next: Next

Picture1.Cls
Picture1.PaintPicture LoadPicture(CurDir$ + "¥earth.gif"), 0, Picture1.Height - 70, 780,
68
Refresh
Label1.Caption = "HI-SCORE " + CStr(HSCR) + "    SCORE 0 "
MTSU = 0: '隕石数
MLSU = 0: 'ミサイル数
For J = 1 To 7: BK(J) = 0: Next
For J = 1 To 3: SX(J) = 0: Next
Timer1.Interval = 200: Timer2.Interval = 50
If SCR >= 20000 Then
    MTSU = 4: MLSU = 4
    BX(4) = 0: BY(4) = 80: BK(4) = 21
    Timer2.Interval = 0
End If
SCR = 0: TM = 0: MMAX = 4

End Sub

Private Sub PUTA()

Dim M1 As Integer, N1 As Integer, Z As Integer

If K <> 80 Then
    If M < 2 Then M = 2
    If M > 252 Then M = 252
End If
M1 = 3 * M - 3: N1 = 3 * N - 3

Select Case K Mod 100
    'meteo
    Case 3, 8, 11, 14, 16
        Z = BitBlt(Picture1.hDC, M1, N1, 9, 9, Picture2.hDC, 0, 0, &HCC0020)
        For N1 = N To N + 3: For M1 = M To M + 3: A(M1, N1) = K: Next: Next
    Case 2, 4, 5, 7, 10, 13
        Z = BitBlt(Picture1.hDC, M1, N1, 15, 15, Picture2.hDC, 12, 0, &HCC0020)
        For N1 = N To N + 4: For M1 = M To M + 4: A(M1, N1) = K: Next: Next
    Case 1, 6, 9, 12, 15, 17
        Z = BitBlt(Picture1.hDC, M1, N1, 24, 24, Picture2.hDC, 27, 0, &HCC0020)
        For N1 = N To N + 7: For M1 = M To M + 7: A(M1, N1) = K: Next: Next
    'ｽﾍﾟｰｽｼｬﾄﾙ
    Case 21
        Z = BitBlt(Picture1.hDC, M1, N1, 45, 15, Picture2.hDC, 56, 0, &HCC0020)
    'ミサイル
    Case 80
        If N < 146 Then
            Z = BitBlt(Picture1.hDC, M1 - 1, N1, 5, 24, Picture2.hDC, 51, 0, &HCC0020)
        End If
End Select

End Sub
Private Sub PUTB()

Dim M1 As Integer, N1 As Integer, Z As Integer
'消す
M1 = 3 * M - 3: N1 = 3 * N - 3
Select Case K
    Case 3, 8, 11, 14, 16
        Z = BitBlt(Picture1.hDC, M1, N1, 9, 9, Picture2.hDC, 91, 0, &HCC0020)
        For N1 = N To N + 3: For M1 = M To M + 3: A(M1, N1) = 0: Next: Next
    Case 2, 4, 5, 7, 10, 13
        Z = BitBlt(Picture1.hDC, M1, N1, 15, 15, Picture2.hDC, 91, 0, &HCC0020)
        For N1 = N To N + 4: For M1 = M To M + 4: A(M1, N1) = 0: Next: Next
    Case 1, 6, 9, 12, 15, 17, 21
        Z = BitBlt(Picture1.hDC, M1, N1, 24, 24, Picture2.hDC, 91, 0, &HCC0020)
        For N1 = N To N + 7: For M1 = M To M + 7: A(M1, N1) = 0: Next: Next
    Case 80
        If N < 146 Then
            Z = BitBlt(Picture1.hDC, M1 - 1, N1, 5, 24, Picture2.hDC, 91, 0, &HCC0020)
        End If
End Select

End Sub


Private Sub Form_Load()

Form1.Left = 0: Form1.Top = 0
Form1.Width = Screen.Width: Form1.Height = Screen.Height - 150
Picture1.Width = 780: Picture1.Height = 525
Picture1.Left = (Form1.ScaleWidth - Picture1.Width) ¥ 2
Picture1.Top = (Form1.ScaleHeight - Picture1.Height) ¥ 2
Picture1.BackColor = &H0: Picture1.ForeColor = &HFFFFFF
Picture1.Font.Size = 24
Label1.Top = Picture1.Top - 20
Call NEWGAME

End Sub



Private Sub Picture1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)

Dim J As Integer, Z As Integer
```


★「今あるデータはいずれ消える…何とかしないと」「…石に記録すれば？ミゾほってさ」「…矢追だね？」（愛知県名古屋市・**ズオウ・リアキ29歳**）……【**つぐ美**：矢追って，矢追純一のことかなぁ？**編**：うーん？？**影**：だから誰と誰が会話してるの？！これは！！】

```
If MLSU > 2 Then Exit Sub
For J = 1 To 3
    If SX(J) = 0 Then
        Z = SNDPLAYSOUND(CurDir$ + "¥hyu.wav", 1)
        MLSU = MLSU + 1: SX(J) = X ¥ 3 + 1: SY(J) = 153
        Exit For
    End If
Next

End Sub

Private Sub TIMER1_Timer()

Dim F As Integer, J As Integer, Z As Integer
Static SD As Integer, T As Integer, X1 As Integer

Randomize

F = 0
For J = 1 To MMAX
    M = BX(J): N = BY(J): K = BK(J)
    If K > 0 Then Call PUTB
    Select Case K
        Case 0
            N = 0
        Case 1, 2, 3, 15, 17
            N = N + SD + 1
        Case 4, 5
            If BK(J) = 4 Then M = M + SD Else M = M - SD
            N = N + SD
        Case 6, 7, 8
            M = M + 1 + 2 * (Int(Rnd * 2) = 0): N = N + SD
        Case 9, 10, 11
            M = M + 2 + 4 * (Int(Rnd * 2) = 0): N = N + SD
        Case 12, 13, 14
            T = T - 1
            If T < 1 Then
                T = 40: If X1 = 1 Then X1 = -1 Else X1 = 1
            End If
            M = M + 3 * X1: N = N + 2
        Case 16
            M = M + 1 + 6 * (N Mod 2 = 0): N = N + 1
            If N > 80 Then BK(J) = 14
        Case 21
            M = M + 2
            Picture1.CurrentY = 230
            If M > 60 And M < 220 And M Mod 20 = 0 Then
                Picture1.CurrentX = 60 * (M ¥ 20) - 30
                Picture1.Print Mid("TOAMKENDUENTAORIN2002", 2 * (M ¥ 20) - 6, 1)
            End If
            If M > 250 Then Call NEWGAME: Exit Sub
    End Select
    If K > 0 Then K = 100 * J + BK(J): Call PUTA: BX(J) = M: BY(J) = N
    If N > 152 Then F = 1: Exit For
Next
'隕石発生
If MTSU < MMAX And Int(Rnd * 4) = 0 Then
    For J = 1 To MMAX
        If BK(J) = 0 Then Exit For
    Next
    M = Int(Rnd * 250) + 2: N = 0
    If J < 3 Then
        Select Case TM
            Case 0 To 17
                K = TM ¥ 6 + 1
            Case 18 To 32
                K = Int(Rnd * 9) + 6
                If M < 90 And Int(Rnd * 2) = 0 Then
                    K = 4
                ElseIf M > 180 And Int(Rnd * 2) = 0 Then
                    K = 5
                End If
            Case Else
                K = Int(Rnd * 17) + 1
        End Select
    Else
        TM = TM + 1
        Select Case TM
            Case 0 To 5
                SD = 1
                If Int(Rnd * 2) = 0 Then
                    K = 4: M = Int(Rnd * 80) + 2
                Else
                    K = 5: M = Int(Rnd * 60) + 170
                End If
            Case 6 To 23
                MMAX = 5
                If TM < 12 Then SD = 2 Else SD = 3
                M = Int(Rnd * 40) + 100
                K = TM ¥ 2 + 3
            Case 24 To 26
                MMAX = 6
                K = 16: M = Int(Rnd * 40) + 210
            Case 27 To 32
                Do
                    M = Int(Rnd * 250) + 2
                Loop While (M > 80 And M < 170)
                If TM < 29 Then K = 15 Else K = 17
            Case Else
                K = Int(Rnd * 3) + 15
        End Select
    End If
    MTSU = MTSU + 1: BX(J) = M: BY(J) = 0: BK(J) = K
    K = 100 * J + BK(J)
```

★春ですね。花粉の季節です。今年になって「薬で予防する」ということを覚えました。我ながら遅すぎ…。(東京都中央区・**まぢけん20歳**)……**【つぐ美：**緑黄色野菜や食物繊維の多い豆・海藻・キノコ類を食べるといいらしいわよ。もう花粉の季節も終わったかな？**】**

```
    Call PUTA
End If
Picture1.Refresh
If F = 1 Then
    Z = SNDPLAYSOUND(CurDir$ + "¥ERT.WAV", 1)
    MsgBox "人類は滅亡した!!", 0 + 48, "ｹﾞｰﾑ ｵｰﾊﾞｰ"
    If SCR >= 20000 Then
        MsgBox "でも、２００００点達成!!", 0 + 48, "やったね"
    Else
        MsgBox "20000点以上でｽﾍﾟｰｽｼｬﾄﾙが見れる!!", 0 + 48, "ｹﾞｰﾑ ｵｰﾊﾞｰ"
    End If
    SD = 1: Call NEWGAME
End If


End Sub

Private Sub TIMER2_Timer()

Dim F As Integer, J As Integer, P As Integer, T As Integer, Z As Integer
'ミサイル飛ぶ
If MLSU = 0 Then Exit Sub
For J = 1 To 3
    M = SX(J): N = SY(J)
    If M > 0 Then
        K = 80: Call PUTB
        F = 0
        For T = 0 To 3
            N = N - T: SY(J) = N
            If N < 1 Then SX(J) = 0: MLSU = MLSU - 1: Exit For
            If A(M, N) <> 0 Then
                F = A(M, N) ¥ 100: N = 0
                SX(J) = 0: MLSU = MLSU - 1
                Exit For
            ElseIf A(M, N + 2) <> 0 Then
                F = A(M, N + 2) ¥ 100: N = 0
                SX(J) = 0: MLSU = MLSU - 1
                Exit For
            End If
        Next
        If N > 0 Then Call PUTA
        If F > 0 Then
            Z = SNDPLAYSOUND(CurDir$ + "¥ban.wav", 1)
            M = BX(F): N = BY(F): K = BK(F): Call PUTB
            If K = 15 Or K = 17 Then
                If K = 15 Then
                    K = 4 - (M > 100)
                    M = M + 5 + 10 * (K = 5)
                    If M < 2 Then M = 2
                    If M > 252 Then M = 252
                    BX(F) = M
                Else
                    K = 13
                    If MTSU < MMAX Then
                        MTSU = MTSU + 1
                        For P = 1 To MMAX
                            If BK(P) = 0 Then Exit For
                        Next
                        M = M - 5: If M < 2 Then M = 2
                        BX(P) = M: BY(P) = BY(F): BK(P) = 5
                        K = 4: M = M + 5
                        If M > 252 Then M = 252
                        BX(F) = M
                    End If
                End If
                BK(F) = K
                K = 100 * F + BK(F): Call PUTA
            Else
                BK(F) = 0: MTSU = MTSU - 1
            End If
            SCR = SCR + 2 * (160 - N)
            If SCR > HSCR Then HSCR = SCR
            Label1.Caption = "HI-SCORE " + CStr(HSCR) + "      SCORE " + CStr(SCR) + " "
        End If
    End If
Next
Picture1.Refresh

End Sub
```

《リスト２》モジュールのプログラム・リスト

```
Attribute VB_Name = "Module1"
Option Explicit

Declare Function SNDPLAYSOUND% Lib "winmm" Alias "sndPlaySoundA" (ByVal Fn$, ByVal Flag%)
Declare Function BitBlt% Lib "gdi32" (ByVal destDC%, ByVal X%, ByVal Y%, ByVal w%, ByVal
H%, ByVal srcDC%, ByVal xSrc%, ByVal ySrc%, ByVal Rop&)

Public A(261, 164) As Integer
Public K As Integer, M As Integer, N As Integer
Public TM As Integer, MMAX As Integer, MTSU As Integer, MLSU As Integer
Public SCR As Long, HSCR As Long
Public BX(7) As Integer, BY(7) As Integer, BK(7) As Integer
Public SX(3) As Integer, SY(3) As Integer
```

★ベーマガ買い始めて10年。本の収納場所に困っているけど，OFネタにプログラム，ちょっと楽しい。P.S.今度プログラム出します。
（群馬県高崎市・巽　洋平22歳）……【影：楽しくなけりゃベーマガじゃない！多くの人に楽しんでもらえるよう，がんばります！】

Windows(Visual Basic V6.0 + DirectX)用



架空の乗り物でレースゲーム

BIKE(バイク)

橘　一馬（30歳）

はじめに

「ドラゴンボール」を読んでいると，タイヤのないスクーターみたいのがたまに出てくる。それを見ていて「こういうバイクに乗りたい！」って思って，こういうゲームができました。

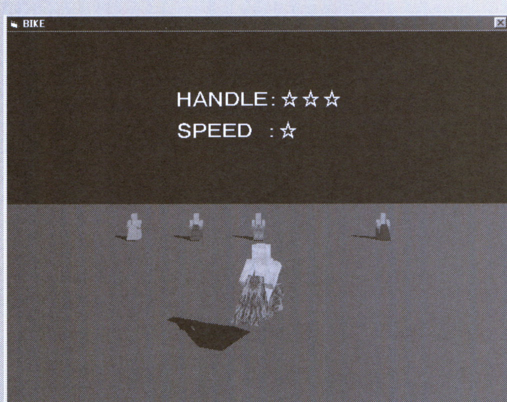
僕がゲームを作る時は，大体こんな感じでゲームを作ります。今度はどんなの作ろうかな〜。

遊び方

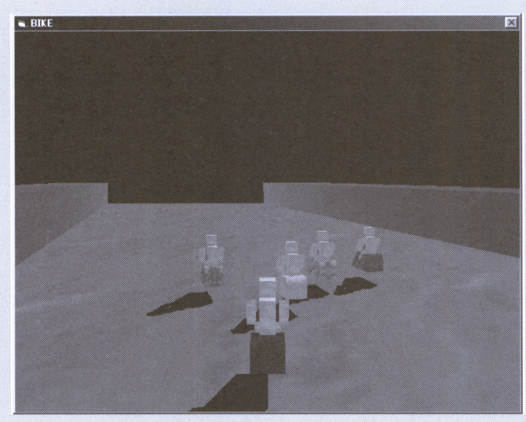
キーボードか，ジョイスティックで遊びます。

タイトル画面でスペースキーを押した後，バイク選択画面で好きなバイクを選びます(**写真１**)。バイクの決定は，↑もしくは↓キーです。バイク選択後，レース画面になります。

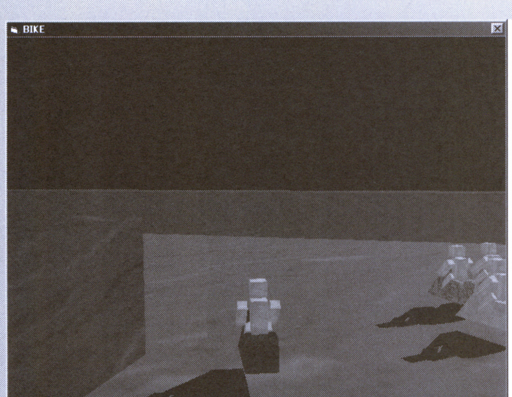
レース中は，カーソルキーの左右←→でハンドル，スペースキーでアクセル操作となります(**写真２，３**)。敵のバイク



《写真１》オリジナリティのあるバイクを使用



《写真２》完全な3D形式のレースゲーム



《写真３》コーナリングが勝負のポイント



《写真４》これがゴールした画面。１着を目指せ

より早くゴールに行ってください(**写真4**)。

ジョイスティックの場合は，方向キーと1キーで，キーボードと同じように操作します。

改造

ステージは，今のところ全5面です。ステージマップは「Map?.txt」というファイルで記述されています。誰でも手軽に変更できるようにテキストファイルにしてます。ステージ数を増やしたり，コース内容を改造したりして遊んでください。ステージデータは，以下のようになっています。

0：下，1：右下，2：右，3：右上，4：上，5：左上，6：左，7：左下，8：壁，9：ゴール。0から7はCPUの動きを示します。

マップは適当に作ったのですが，それによってゲームの面白さが全然変わってくるのだな～と感じました。面白いゲームって,やっぱりこういうところもよくできているな～と思いました。特にパズルゲームとかシューティングゲームとか。

カーレースゲームは，いろいろと知識が必要なんだな～と思いました。ドリフトをしているつもりなんですがうまくいきません。壁との当たり判定もよくありません。もっといいアルゴリズムがあれば教えてほしいのですが…。

いつもプログラミングしていて気になるのは，違う環境で動くかって事です(特に3Dは)。しかし，僕のパソコンは1台なので，他の環境でテストができません。なので，他のパソコンでどのように動くか教えてくれるとうれしいです。

参考にしたもの

・音・辞典(TM) VOL.1［アニメ・ゲーム効果音］

・Metasequoia

・Visual BasicによるDirectX7活用術　川田徹

・DirectX7 For Visual Basic　はじめるゲームプログラミング　藤田伸二

CHECKER FLAG

つぐ美：3Dレースゲームとして見れば，ホントに基本的な内容でチョット物足りないのですが，プログラム面から見ると，いろいろと参考になるところが多いんじゃないかしら？

Dr.D：やはり3Dということで，コースやバイクといったモデルの表示や，カメラの移動処理などに注目してリストを解析してみると良いのではないかな？この部分の処理がわかれば，ゲーム自体のだいたいの処理も，おのずと理解できてくるはずじゃ。

編：余談ですが，何かの本にこういったタイヤのない車は，タイヤの摩擦が発生しないため，坂道での発進や走行が難しいみたいなことが書かれていた記憶があります。

Pick Up LIST

《リスト》カメラの位置設定処理のリスト

```
Private Sub CameraSyori()
    With Camera
'通常
        .X = Bike(PlayerNo).X - Sin(Radian(Bike(PlayerNo).Ry2)) * 12
        .Z = Bike(PlayerNo).Z - Cos(Radian(Bike(PlayerNo).Ry2)) * 12
        .Y = 5
        .Vx = 0
        .Vy = 0
        .Vz = 0
        .Rx = 5
        .Ry = Bike(PlayerNo).Ry2
        .Rz = 0
        .RVy = 0
        .RVz = 0

'以下はコメント化されており，実際には使用されていませんが，
'参考のため，掲載してあります。

'上
'        .X = Bike(PlayerNo).X
'        .Z = Bike(PlayerNo).Z
'        .Y = 50
'        .Vx = 0
'        .Vy = 0
'        .Vz = 0
'        .Rx = 90
'        .Ry = 0
'        .Rz = 0
'        .RVy = 0
'        .RVz = 0

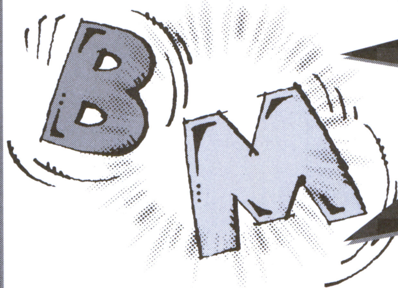
        CameraSet
    End With
End Sub
```

Dr.Dのワンポイント解説

Dr.D：CHECKER FLAGでも触れたカメラ処理部分のリスト。実際に見てみると，行っている処理は，変数の値の設定だけなのじゃが，カメラ関連の変数にどのような値を設定するか，という点がカメラ処理では重要になってくるのじゃ。要は，テレビカメラワークと同じで，どの位置からどの方向へカメラを置けば意図どおりの画面になるか，という思考ができるかが大切なのじゃ。

★定期購読始めました。生協の定期購読です。10%引きでとってもお得。／生協でADSLをたのみました。快適インターネット生活まであとわずか。／1人暮らしの引っ越しで一番きつかったもの，それはベーマガの輸送。(愛媛県松山市・**毛利祐允18歳**)……【**影：う。**】

Windows(VBScript + DirectX)用



3D文字で動体視力判定

くるくるアイ

金井　義男(39歳)

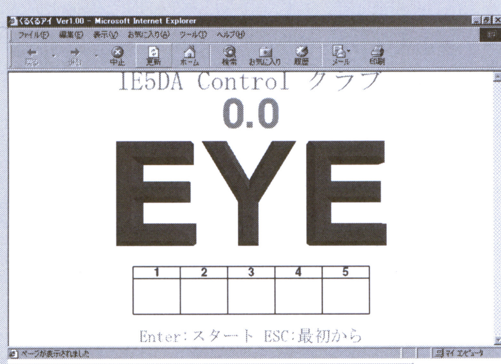
内容

DirectAnimationによる動体視力判定を行います。インチキっぽいけど…。

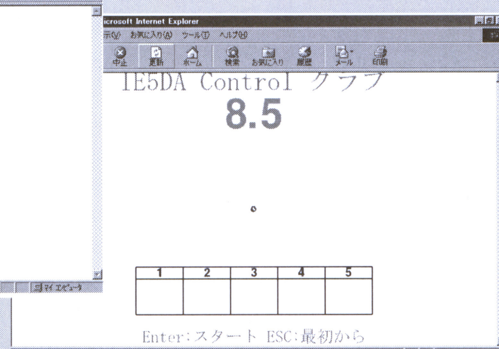
使い方・遊び方

Enterキーでゲームを開始します。アルファベット文字が回転しながら拡大してくるので，文字がわかった時点で，キーボードからその文字を入力してください(**写真2～4**)。

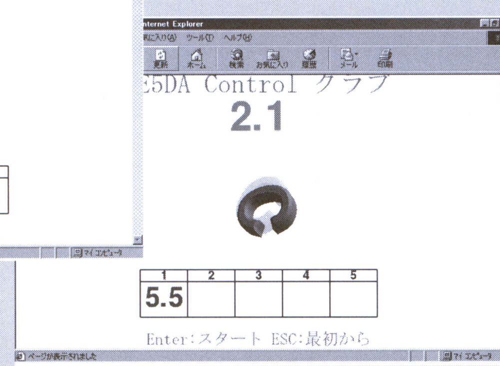
5回で終了し，平均値(動体視力)が表示されます(**写真5**)。再挑戦はEsc



《写真1》3D文字を使った簡易動体視力検査



《写真2》画面奥から回転して現れる文字を判別



《写真3》ここまで接近すれば，もうわかるよね？



《写真4》測定を5回繰り返して行えば…



《写真5》最終結果が表示される

キーを押してください。

プログラムの入力方法

メモ帳から入力できますが，もうちょっと便利なテキストエディタを使ったほうがよいと思います。

あとがき

文字の立体が手軽に作れるのは，Direct Animationの利点ですね。本誌2001年12月号のブラインドタッチ・ゲームを参考に作りました。さすがプロ(平井さん)ですな…と感心しました。

安易な作りですが，もうちょっとDirectAnimationをやっていたいと思います。私のホームページでも作品を紹介していますので，暇のある方は見に来てください。ホームページアドレスは，http://www.d7.dion.ne.jp/~kanaisun です。

本誌で紹介している，HSPの動きの速さには感心しています。

CHECKER FLAG

つぐ美：3Dフォントをうまく利用したプログラムですね。なかなか面白いアイデアだわ。

編：画面奥から回転して接近してくるアニメーションも，かなりの出来ですが，文字の色も合わせて変化させている点にも注目してほしいですね。

Dr.D：編クンが述べているアニメーションの他，キー入力処理やタイマー処理もしっかり記述されており，DirectAnimation作品では，かなり完成されたプログラムと言えるじゃろう。ブラウザ(IE)があれば遊ぶことができるので，プログラムにあまり興味がない人にも，ぜひ遊んでもらいたい作品じゃな。

影：ゲーム面でコメントすると，IやL，Vといった，いわば棒の組合せで出来ている文字の判別が意外と難しいゾ。

《リスト》くるくるアイのプログラム・リスト

```
<html>
<head>
<title> くるくるアイ Ver1.00 </title>
</head>
<body>
<SCRIPT LANGUAGE="VBScript">
'――――――――――――――――――――――――――――――――――――
'// グローバル変数の宣言
'――――――――――――――――――――――――――――――――――――
Option Explicit
Dim m                'As DAStatics
Dim text3d'As DAString      '// 出題文字イメージ
Dim geo        'As DAGeometry'// 出題文字３Dイメージ
Dim newgeo     'As DAGeometry'// ３D動作設定イメージ
Dim bkimg      'As DAImage   '// ２D背景イメージ
Dim timeimg     'As DAImage   '// １０．０→０．０カウントダウンイメージ
Dim newtimeimg      'As DAImage        '// カウントダウンリフレッシュ用
Dim cntimg(4)       'As DAImage        '// ステージタイムイメージ
Dim timecnt         'As Long  '// ゲーム用タイムカウンター
Dim penaltycnt      'As Long  '// ペナルティー用タイムカウンター
Dim codeno'As Long  '// 出題文字キーコードNO
Dim cntno 'As Long  '// ステージNOカウンター
Dim flag  'As Boolean         '// ゲームスタートフラグ
Dim penalty         'As Boolean        '// ペナルティフラグ
Dim total 'As Single'// クリアタイムの合計
Const LIMIT = 10              '// 制限時間(秒)
Const LINEL = 0.02            '// １辺の長さ（イメージ配置基準値）

'――――――――――――――――――――――――――――――――――――
'// 初期化処理
'――――――――――――――――――――――――――――――――――――
Sub Window_onLoad()
'//変数の宣言
Dim light      'As DAGeometry'// 光源ライトイメージ
Dim camera     'As DACamera  '// カメラ設定
Dim geoimg     'As DAImage   '// レンダリングイメージ
Dim hsl        'As DAImage   '// 変化するライト色イメージ
Dim num             'As DANumber       '// ライトの動作設定値
Dim evt        'As DAEvent   '// タイマー用イベント
Dim none       'As DASound    '// イベントハンドリング

 '// 乱数ジェネレータ の初期化
 Randomize
 cntno = 0
 '// DirectAnimationの初期化 (メートル座標系)
 Set m = DAControl.MeterLibrary
 '// 文字を差し替えるために、ビヘイビアで定義
 Set text3d = m.ModifiableBehavior((m.DAString("EYE")))
 '// 文字の動作を差し替えるために、ビヘイビアで定義
 Set geo = m.ModifiableBehavior(m.EmptyGeometry)
 '// 立体文字作成と動作設定
 Call MojiMake()
 '// タイマー表示を差し替えるために、ビヘイビアで定義
 Set timeimg = m.ModifiableBehavior(m.EmptyImage)
 '// 画面背景の作成
 Call Field_Make()
 '// 光源の設置 (ポイント ライト)
 Set num = m.Mul(m.LocalTime, m.DANumber(0.05))
 Set hsl = m.ColorHslAnim(num, m.DANumber(1), m.DANumber(0.5))
 Set light = m.PointLight.LightColor(hsl)
 Set light = light.Transform(m.Translate3(1, 1, 1))
 '// ライトと物体を合成
 Set geoimg = m.UnionGeometry(geo, light)
 '// レンダリング (パースペクティブ カメラ)
 Set camera = m.PerspectiveCamera(0.2, 0.19)
 Set geoimg = geoimg.Render(camera)
 '// タイマーイベント
 Set evt = m.Timer(0.1)
 Set evt = evt.ScriptCallback("NextTime", "VBScript")
 Set none = CreateObject("DirectAnimation.DASound")
 none.Init m.Until(m.Silence, evt, none)
 '// コントロールにオブジェクトをセット
 DAControl.Image = m.OverlayArray(Array(timeimg, bkimg, geoimg))
```

★本棚を整理していたら83年8月のベーマガが出てきました。思わず読みふけってしまいました。(神奈川県横浜市・よこくる23歳)…
…【編：ある，ある。掃除中や整頓中に，懐かしい物を見つけると，読みふけてしまって，掃除どころじゃなくなるってこと。】

```
  '// アニメーション開始
  DAControl.Start
  '// DAControlのSoundプロパティにセットする
  DAControl.Sound = none
  Call Window_OnResize
End Sub

'---------------------------------------
'// 画面サイズ変更時の再描画
'---------------------------------------
Sub Window_OnResize()
On Error Resume Next
  DAControl.Width  = Document.Body.ClientWidth
  DAControl.Height = Document.Body.ClientHeight
  DAControl.focus
End Sub

'// ---------------------------------------
'NextTime : タイマーカウンター
'// ---------------------------------------
Private Sub NextTime()
  Dim newimg         'As DAImage         '// ステージタイム差し替えイメージ
  Dim fnt 'As DAFontStyle '// フォントスタイル
  Dim i              'As Long  '// ループカウンター

  If flag = True Then
    If penalty = True Then
      '文字を間違えた場合、1秒間ペナルティ フラグを立てる
      penaltycnt = penaltycnt + 0.1
        If penaltycnt > 1 Then penalty = False
    End If
    timecnt = timecnt + 0.1
      If timecnt > LIMIT And cntno < 5 Then
          Set fnt = m.Font("Tahoma", 32, m.Blue).Bold
          Set newimg = m.StringImage("0.0", fnt)
          cntimg(cntno).SwitchTo newimg
        flag = False
          cntno = cntno + 1
            If cntno = 5 Then
             '結果の表示
             Call End_Mess
            End If
      End If
  End If

End Sub

'//-------------------------------------
'// 立体文字を作成し、動作設定を定義する
'//-------------------------------------
Private Sub MojiMake()
Dim dxt             'As Object'// DXTMSFT3Lib.DXTText3D
Dim res         'As  IDADXTransformResult
Dim mesh1,mesh2 'As  DAGeometry '// ３Ｄ文字物体
Dim tmp             'As DAGeometry      '// 一時保存用
Dim num             'As DANumber        '// 文字拡大動作設定値

  '// DirectX Transform オブジェクトの作成 (Text3D)
  Set dxt = CreateObject("DX3DTransform.Microsoft.Text3D")
  '// DirectX Transform の適用
  Set res = m.ApplyDXTransform(dxt, Null, Null)
  '// DirectX Transform のプロパティ設定
  res.PutBvrAsProperty "FontFace", m.DAString("Arial")
  res.PutBvrAsProperty "String", text3d
  res.PutBvrAsProperty "ExtrusionType", m.DAString("FlatBevel")
  '// トランスフォーム適用後のジオメトリを取得
  Set mesh1 = res.OutputBvr
  '文字の中心を座標の原点へ移動
  Set mesh1 = mesh1.Transform(m.Translate3(0, -0.25, 0))
  '// 文字を徐々に拡大
  Set num = m.Sequence(m.SlowInSlowOut(LINEL / 10, LINEL / 2, LIMIT * 0.6, 1), _
                              m.SlowInSlowOut(LINEL / 2, LINEL * 3, LIMIT * 0.4, 1))
  Set mesh2 = mesh1.Transform(m.Scale3UniformAnim(num))
  '// Z軸回転(1秒で3回転)
  Set newgeo = mesh2.Transform(m.Rotate3RateDegrees(m.ZVector3, 360 * 3))
  '// 高速で回転している様にみせるための残像用物体の定義(60度ずらす)
  Set tmp = mesh2.Transform(m.Rotate3Degrees(m.ZVector3, 60))
  '// タイミングをずらした上でZ軸回転(1秒で3回転)
  Set tmp = tmp.Transform(m.Rotate3RateDegrees(m.ZVector3, 360 * 3))
  '// 残像用物体は徐々に透明にする
  Set tmp = tmp.OpacityAnim(m.Interpolate(1, 0, LIMIT))
  '// ２つの物体を合成
  Set newgeo = m.UnionGeometry(tmp, newgeo)
  '// さらに、X軸とY軸の回転を与える
  Set newgeo = newgeo.Transform(m.Rotate3RateDegrees(m.XVector3, 360))
  Set newgeo = newgeo.Transform(m.Rotate3RateDegrees(m.YVector3, 360))
  '// 拡大後の最終イメージ
  Set mesh1 = mesh1.Transform(m.Scale3Uniform(LINEL * 3))
  '// １０秒間geo その後 mesh1
  Set newgeo = m.Sequence(newgeo.Duration(LIMIT), mesh1)
  '// 物体のスペキュラー(反射光) を設定
  Set newgeo = newgeo.SpecularExponent(10)
  geo.SwitchTo newgeo
End Sub

'//-------------------------------------
'// タイマーとスコアーボード(１～５)を定義する
'//-------------------------------------
Private Sub Field_Make()
  '変数・定数の宣言
  Dim line'As DAPath2          '// 線描画設定
  Dim border    'As DAImage    '// 線の描画プロパティ
  Dim tmpimg    'As DAImage    '// 一時保存用
  Dim scoreimg  'As DAImage    '// スコアーを記録する枠イメージ
  Dim tmrStr          'As DAString    '// 制限時間(DAString)
  Dim fnt 'As DAFontStyle '// フォントスタイル
  Dim i               'As Long
  Const LINEW = 1.5            '// 線の幅

  '// 四角形軌跡線の描画
  Set border = m.DefaultLineStyle.Color(m.BLACK).Width(LINEW)
  Set line = m.Rect(LINEL, LINEL)
  Set scoreimg = line.Draw(border)
  '// 直線を引く
  Set line = m.Line(m.Point2(LINEL / -2, LINEL / 4), m.Point2(LINEL / 2, LINEL / 4))
  Set tmpimg = line.Draw(border)
  Set scoreimg = m.Overlay(tmpimg, scoreimg)
```

★只今布教活動ニツキ，4月号からクラス内で寄付金集めてBM買ってます。どうですか？影教祖様。(愛知県一宮市・**幻幹部19歳**)……
【影：なに40人クラスとして，1人13.75円で回し読みか？いやわかっておる，もちろんクラス全員1冊ずつ買っておることは。】

```
'// フォントの定義
Set fnt = m.Font("Tahoma", 14, m.Black).Bold
Set bkimg = m.EmptyImage
For i = 0 To 4
  '// タイム表示イメージをビヘイビアで定義（最初は空の状態）
  Set cntimg(i) = m.ModifiableBehavior(m.EmptyImage)
  '// ステージ番号の定義
  Set tmpimg = m.StringImage(i + 1, fnt).Transform(m.Translate2(0, LINEL / 3.4))
  Set tmpimg = m.Overlay(tmpimg, scoreimg).Transform(m.Translate2(0, LINEL / 3.5))
  Set tmpimg = m.Overlay(tmpimg, cntimg(i)).Transform(m.Translate2(LINEL * i - LINEL * 2,
LINEL * -2))
  Set bkimg = m.Overlay(tmpimg, bkimg)
Next
'// メッセージイメージの作成
Set fnt = m.DefaultFont.Color(m.ColorRgb255(125, 125, 0))
Set fnt = fnt.Family("ＭＳ 明朝").Size(32).Bold
Set tmpimg = m.StringImage("IE5DA Control クラブ", fnt)
Set tmpimg = tmpimg.Transform(m.Translate2(0, LINEL * 2.5))
Set bkimg = m.Overlay(tmpimg, bkimg)
Set fnt = fnt.Family("ＭＳ 明朝").Size(20).Bold
Set tmpimg = m.StringImage("Enter:スタート ESC:最初から", fnt)
Set tmpimg = tmpimg.Transform(m.Translate2(0, LINEL * -2.8))
Set bkimg = m.Overlay(tmpimg, bkimg)
'// タイマーイメージの作成
Set fnt = m.Font("Tahoma", 48, m.Red).Bold
'// 動的なDAStringオブジェクトからDAImageオブジェクトへ
Set newtimeimg = m.StringImageAnim(m.Interpolate(LIMIT, 0, LIMIT).ToString(1), fnt)
Set newtimeimg = newtimeimg.Transform(m.Translate2(0, LINEL * 1.7))
timeimg.SwitchTo newtimeimg
End Sub

'//---------------------------------------------
'// コントロールのKeyDownイベントプロシージャ
'//---------------------------------------------
Private Sub DAControl_KeyDown(ByVal KeyCode, ByVal KeyData)
  Dim fnt       'As DAFontStyle  '// フォントスタイル
  Dim newimg         'As DAImage

  '// KeyUpイベントを無視する
  If KeyData < 0 Then Exit Sub
  '// ペナルティを受けている間は、キー入力を無視する
  If penalty = True Then Exit Sub
  '//キーコードによる分岐
  Select Case KeyCode
    Case 13   '[Enter]
      If flag = False Then
          '// 65 から 90 までの乱数を発生させる
          codeno = Int((26 * Rnd) + 65)
          '// KeyCodeで文字作成し、差し替える
          text3d.SwitchTo m.DAString(Chr(codeno))
          '// 文字の動作をリフレッシュ
          geo.SwitchTo newgeo
          '// タイマーの動作をリフレッシュ
          timeimg.SwitchTo newtimeimg
          '// ゲームスタートフラグ
          flag = True
          timecnt = 0
      End If
    Case 27   '[ESC]
      Window.Navigate("eyemove.html")    '← リフレッシュ
    Case KeyCode < 65 Or KeyCode > 90
      Exit Sub
    Case Else
      If flag = True Then
          If KeyCode = codeno Then
          '// 正解タイムの表示
            Set fnt = m.Font("Tahoma", 32, m.Blue).Bold
            total = total + LIMIT - timecnt
            Set newimg = m.StringImage(Left(LIMIT - timecnt, 3), fnt)
            cntimg(cntno).SwitchTo newimg
            cntno = cntno + 1
              If cntno < 5 Then
                flag = False
              Else
                '結果の表示
                Call End_Mess
              End If
        Else
          '// ミスした場合、ペナルティを科す
          penalty = True
          penaltycnt = 0
          End If
      End If
  End Select
End Sub

'//---------------------------------------------
'// 終了結果、平均値の表示
'//---------------------------------------------
Private Sub End_Mess()
  Dim fnt 'As DAFontStyle     '// フォントスタイル
  Dim newimg         'As DAImage          '// 差し替え用イメージ

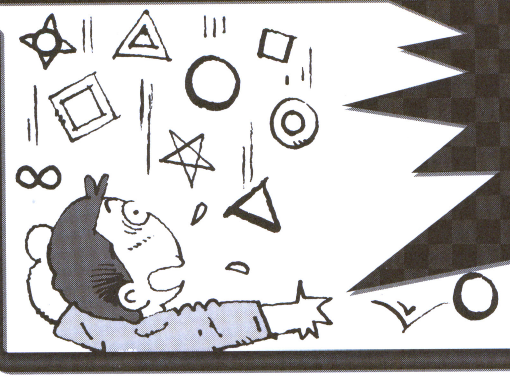
  Set fnt = m.Font("ＭＳ 明朝", 32, m.Blue).Bold
  Set newimg = m.StringImage("動体視力 = " & Left(total / 5, 3), fnt)
  Set newimg = newimg.Transform(m.Translate2(0, LINEL * 1.7))
  timeimg.SwitchTo newimg
End Sub
</SCRIPT>

<! DirectAnimation コントロール>
<OBJECT ID="DAControl"
  CLASSID="CLSID:B6FFC24C-7E13-11D0-9B47-00C04FC2F51D"
  STYLE="position:absolute:left:0:top:0">
</OBJECT>

</body>
</html>
```

★4月号の53ページを見て，なぜか「牛食えを復活させたらどうかな…」とか思ってしまいました。（新潟市・**佐藤良秋21歳**）……
【影：「牛食え」はWeb上で展開中です。そちらもよろしく！あー，牛丼食べたくなってきた。つぐ美ちゃん，ご一緒しません？**】**

Excel(VBA)用



VBAで落ち物パズルを実現

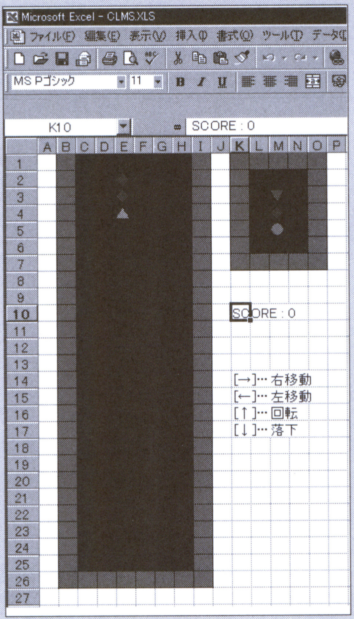
CLMS

年齢不相応（42歳）

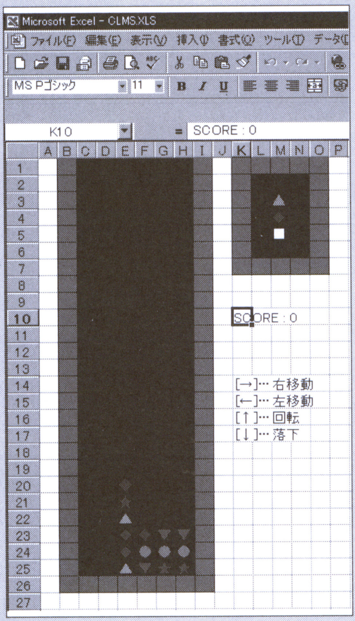
はじめに

ちょっと面白いプログラムができたので投稿させていただきます。いえ…モノ自体はありふれたものなのですが，なんとエクセルのシートとVBAを使って，困難とされていたリアルタイムゲームを実現(大袈裟)したのです。

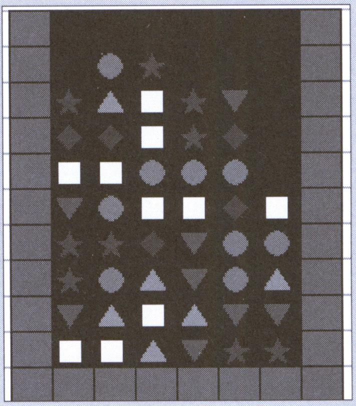
まあ，ゲームの内容はともかく，世にも珍しいエクセルのリアルタイムゲーム，お楽しみください。



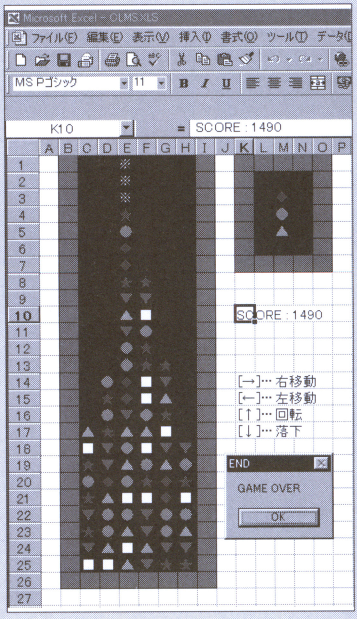
《写真１》Excel上で落ち物パズルが遊べる



《写真２》某有名な落ち物パズルと同じルール



《写真３》連鎖を組んで高得点を叩き出せ



《写真４》画面上まで積み上がるとダメ

★職場のPCで，ウイルスのバッドトランスBを受信。ウイルスを目の当たりにしたのは，ラルーに続き2度目。駆除ソフトもバージョンが古くて大騒ぎ。身近で起こると他人事ではないと思いました。（島根県松江市・NSR250R＋横浜ナンバー38歳）……【編：はい。】

使い方・遊び方

エクセルシートのTITLEページにあるSTARTボタンを押してください。ゲームがスタートします(**写真１**)。

ゲームのルールは，上から落ちてくるジュエルを操作し，同じ色と形をもつジュエルを縦，横，斜めのいずれかに三つ並ぶようにします(**写真２,３**)。この条件を満たすとジュエルが消え，得点が入ります。ジュエルが上まで積み上がってしまうと，ゲームオーバーです(**写真４**)。

操作方法

カーソルキーの[←][→]で落ちてくる物を操作します。カーソルキーの[↓]を押すと一気に下まで落ちます。カーソルキーの[↑]を押すと，ジュエルの色がシフトします。

また，途中で止めたくなった場合は，[Esc]キーを押してください([Break]キーで止めると，エクセルシート画面では，キー入力にロックがかかったままになってしまいます)。

それと，このゲームはエクセルシートを直接使用しているので，ZOOMの倍率を変える事で，かなり小さく表示できます。これをうまく利用すれば，仕事中にこそこそ遊ぶなんて事も…(笑)。

プログラムの仕組み

上記の内容からわかるとおり，ゲームの内容は多くのアマチュアプログラマーたちの手で何度も再現されたものですし，ロジックもありふれた代物です(ちょっと作りは雑ですが…)。

このプログラムの肝はVBAで擬似リアルタイム入力を可能にしたApplication.OnTimeとApplication.OnKeyの，ふたつの命令です。

VBAには，基本的に(自分の知る限りでは)リアルタイムのキー入力命令が存在しません。そこでApplication.OnTimeで一時的に制御をマクロからエクセルシートに戻し，その間Application.OnKeyを使用して，別のマクロを起動する事で，リアルタイムに処理が行われているように見せかけているのです。

上記のプログラムでは，ジュエルの落下処理をApplication.OnTimeでループさせ，Application.OnKeyでカーソルの上下左右のキーにジュエルの左，右，回転，一気落下の処理を起動させています。

入力方法と注意点

エクセルシートから「ツール」→「マクロ」→「Visual Basic Editor」を選んで，標準モジュールを追加して，掲載されているリスト入力すればOKです。

また，エクセルのシートに「TITLE」「VIEW」「HIDE」と名前を付けてあげてください。

「TITLE」シートへのボタンの貼り付けは「表示」→「ツールバー」→「フォーム」で，最後に起動するマクロをINIT00にしてください。

実際のプレイ画面は「VIEW」になり，「HIDE」は通常のプログラムで言うところの仮想画面(配列等)の役割を果たしています。

また，セルのあちこちに数値を書き込む事で，マクロ同士のデータの受け渡しをやっています。実はプレイ画面にもそういう書き込みがあるのですが，フォントのカラーを背景と同じにすることで誤魔化しています。このへんは「hide」のシートを使えば良かったのですが，開発中にゲーム画面と同時に制御データを監視したかったため，こうしました。

改造のアドバイス

わかる人なら説明するまでもないと思います。プログラムの基本的な構造は，ブロックの表示処理，衝突処理，消去処理ですので，そのへんをいじればオリジナルの落ち物を簡単に作れると思います。

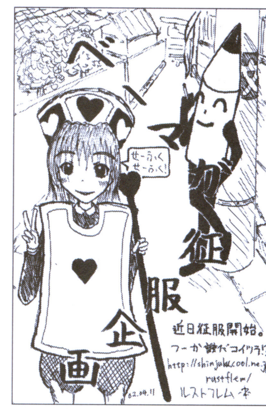
苦労した点

Application.OnTimeで指定できる時間が最小で１秒後のため，ブロックの落下速度を上げるのが困難でした。で，どうしたかというと１秒おきに２個分，３個分とまとめて落としています。おかげで動きがかなりカクカクしていますが，これは現在の自分の技術力では解決できませんでした。



評…どんどん投稿してね(編)

by みなみとおれが



評…かっこいい～制服?!(影)

by ルストフレム

CHECKER FLAG

影：おっ，この作品はOFの常連である年齢不相応さんからの投稿だね。

Dr.D：アイデアはスタンダードなものじゃが，このゲームをVBAで実現した点は評価できる。説明文でも触れているように，リスト面での工夫が光っている作品じゃ。

つぐ美：実は，このパズルの他にも，有名パズルゲームを移植した作品が二つ一緒に投稿されてきたのですが，リスト等の関係から，この『Clms』を掲載させていただきました。他の２作品は，ダウンロードファイルへ収録させてもらいました。

編：実は，このプログラム，＊＊中に密かに制作したものらしいです。密かに制作したにもかかわらず，コメントなどがしっかりしている点が素晴らしいです。

★卒業式に寝ちゃいました（小学1年時の入学式以外すべての入学式と卒業式は全部寝てました）。（新潟県三島郡・フリーゲーマー14歳）……【**編集長**：社会に出たら，仕事中居眠りするのだけはやめなよ。**影**：グー，グー。**編集長**：これ，悪い見本（バコッ）。】

《リスト１》Module1.basのプログラム・リスト

```
'初期化処理
Sub INIT00()
    NM00 = "VIEW"
    NM01 = "HIDE"

    'シートをクリアし、セル幅を変更する
    Sheets(NM00).Activate
    Cells.Select
    Selection.Delete Shift:=xlUp
    Selection.ColumnWidth = 2

    Sheets(NM01).Activate
    Cells.Select
    Selection.Delete Shift:=xlUp
    Selection.ColumnWidth = 2

    '壁を灰色に塗る
    Sheets(NM00).Activate
    Range("B1:B26,B26:I26,I1:I26,K1:O1,K7:O7,K1:K7,O1:O7").Select
    Selection.Font.ColorIndex = 16
    Selection.Interior.ColorIndex = 16
    Selection.Interior.Pattern = xlSolid

    '壁を罫線で区切る
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    With Selection.Borders(xlEdgeLeft)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlEdgeTop)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlInsideVertical)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlInsideHorizontal)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With

    '壁の内側を黒く塗る
    Range("C1:H25,L2:N6").Select
    With Selection.Interior
        .ColorIndex = 1
        .Pattern = xlSolid
    End With

    '壁のセルに当り判定用の値を設定
    For IX00 = 1 To 26
        Sheets(NM01).Cells(IX00, 2) = 10
        Sheets(NM01).Cells(IX00, 9) = 10
    Next
    For IX00 = 2 To 9
        Sheets(NM01).Cells(26, IX00) = 10
    Next

'各キーに対応するマクロを設定
    On Error Resume Next
    For IX00 = 0 To 255
        Application.OnKey Chr(IX00), ""
    Next

    Application.OnKey "{UP}", "UP00"
    Application.OnKey "{DOWN}", "DOWN00"
    Application.OnKey "{RIGHT}", "RIGHT00"
    Application.OnKey "{LEFT}", "LEFT00"
    Application.OnKey "{ESC}", "STOP00"

'説明
    Sheets(NM00).Cells(14, 11) = "[→]…右移動"
    Sheets(NM00).Cells(15, 11) = "[←]…左移動"
    Sheets(NM00).Cells(16, 11) = "[↑]…回転"
    Sheets(NM00).Cells(17, 11) = "[↓]…落下"

'ジュエル情報初期値設定
    Sheets(NM00).Range("A30:E40").Select
    Selection.Font.ColorIndex = 2

    Sheets(NM00).Cells(30, 1) = 0
    Sheets(NM00).Cells(31, 1) = 1
    Sheets(NM00).Cells(31, 2) = 1
    Sheets(NM00).Cells(32, 1) = 0
```

★（4月号68ページ）1810年前にバルセロナ大会があったのかと思ってしまったゾイ。（埼玉県さいたま市・i486dx50　14歳）……
【編：失礼しました。いろんな人から指摘がありました。**つぐ美：**ごめんなさい。影さんも謝りなさいよ。**影：**グ，ググゲー。**】**

```
    Sheets(NM00).Cells(36, 1) = 0
    Sheets(NM00).Cells(37, 1) = 0
    Sheets(NM00).Cells(37, 2) = 200
    Call SCORE00

'新 NEXT ジュエル情報作成
    Call NEXT00
    Cells(10, 11).Select

'オンタイム処理を実行開始
    Call ONTIME00
End Sub
```

《リスト２》Module2.basのプログラム・リスト

```
'ジュエル出現＆落下処理
Sub ONTIME00()
    NM00 = "VIEW"       '主要シート名
    NM01 = "HIDE"       '情報シート名
    FG99 = 0            '処理終了フラグ
    'ジュエル情報取得
    SW00 = Sheets(NM00).Cells(30, 1)
    If SW00 = 0 Then    'ジュエルが非活動状態
        GoSub ONTIME10
    Else                'ジュエルが落下中
        GoSub ONTIME20
    End If

    ST00 = Sheets(NM00).Cells(36, 1)
    If FG99 = 0 And ST00 = 0 Then
        GoSub ONTIME40
    Else
        GoSub ONTIME50
    End If
Exit Sub

'ジュエル出現処理
ONTIME10:
    N000 = Sheets(NM00).Cells(33, 1)
    If Not (N000 = 0) Then
        GoSub ONTIME30
    End If

    'NEXT ジュエル情報取得
    Sheets(NM00).Cells(33, 1) = Sheets(NM00).Cells(32, 1)
    Sheets(NM00).Cells(33, 2) = Sheets(NM00).Cells(32, 2)
    Sheets(NM00).Cells(33, 3) = Sheets(NM00).Cells(32, 3)

    'ジュエル出現初期位置
    XX00 = 5
    YY00 = 1

    '新 NEXT ジュエル情報作成
    Call NEXT00

    '落下速度再設定
    Sheets(NM00).Cells(31, 1) = Sheets(NM00).Cells(31, 2)

    'ジュエル衝突判定（パラメータ３が２）
    FG00 = TET000(XX00, YY00, 2)
    If FG00 = 0 Then
        'ジュエル描画処理（パラメータ３が 1）
        FG00 = TET000(XX00, YY00, 1)
    Else
        'ジュエルクラッシュ処理（パラメータ３が４）
        FG00 = TET000(XX00, YY00, 4)
        FG99 = 1
    End If
    Sheets(NM00).Cells(30, 1) = 1
Return

'ジュエル落下処理
ONTIME20:
    MX00 = Sheets(NM00).Cells(31, 1)
    For IX00 = 1 To MX00
        XX00 = Sheets(NM00).Cells(34, 1)
        YY00 = Sheets(NM00).Cells(35, 1)
        GoSub ONTIME21
    Next
Return

ONTIME21:
    'ジュエル衝突判定（パラメータ３が２）
    FG00 = TET000(XX00, YY00 + 1, 2)
    If FG00 = 0 Then
        'ジュエル消去処理（パラメータ３が０）
        FG00 = TET000(XX00, YY00, 0)
        'ジュエル描画処理（パラメータ３が１）
        FG00 = TET000(XX00, YY00 + 1, 1)
    Else
        'ジュエル非活動宣言
        Sheets(NM00).Cells(30, 1) = 0
        '落下ループ終了
        IX00 = MX00 + 1
    End If
Return

'ジュエル壁化＆消去処理
```

★ファルコムの新作に期待！but，我，金ナシ。ではさようなら。（宮城県仙台市・**たけし16歳**）……【**影**：は，さようなら。「さようなら」って，どうも気勢がそがれる言葉ですね。**つぐ美**：影さん，さようなら，あとよろしく！**影**：ほらね。私もさようなら。】

```
ONTIME30:
    XX00 = Sheets(NM00).Cells(34, 1)
    YY00 = Sheets(NM00).Cells(35, 1)
    'ジュエル壁化処理（パラメータ３が 3 ）
    FG00 = TET000(XX00, YY00, 3)

    '整列壁消去
    FG00 = 0
    SC02 = 10
    While FG00 = 0
        FG00 = 1
        GoSub ONTIME31
        '連鎖した場合、得点ＵＰ
        SC02 = SC02 + 10
    Wend
Return

ONTIME31:
'整列壁チェック
    For IX00 = 1 To 25
        For IX01 = 3 To 8
            TX00 = Val(Sheets(NM01).Cells(IX00, IX01))
            '壁があれば処理
            If Not (TX00 = 0) Then
                GoSub ONTIME32
            End If
        Next
    Next

'整列壁消去
    For IX00 = 1 To 25
        For IX01 = 3 To 8
            TX00 = Val(Sheets(NM01).Cells(IX00, IX01))
            '消去対象があれば処理
            If TX00 < 0 Then
                GoSub ONTIME33
            End If
        Next
    Next

'壁落下処理
    For IX00 = 24 To 1 Step -1
        For IX01 = 3 To 8
            TX00 = Val(Sheets(NM01).Cells(IX00, IX01))
            '壁があれば処理
            If Not (TX00 = 0) Then
                IX02 = IX00
                GoSub ONTIME34
            End If
        Next
    Next
Return

ONTIME32:
    '横並びチェック
    Call LNCHK00(IX01, IX00, -1, 0, 1, 0)

    '縦並びチェック
    Call LNCHK00(IX01, IX00, 0, -1, 0, 1)

    '右下がり並びチェック
    Call LNCHK00(IX01, IX00, -1, -1, 1, 1)

    '右上がり並びチェック
    Call LNCHK00(IX01, IX00, -1, 1, 1, -1)
Return

ONTIME33:
    Sheets(NM00).Cells(IX00, IX01) = ""
    Sheets(NM01).Cells(IX00, IX01) = ""

    SC00 = Sheets(NM00).Cells(37, 1)
    SC01 = Sheets(NM00).Cells(37, 2)
    SC00 = SC00 + SC02
    Sheets(NM00).Cells(37, 1) = SC00
    Call SCORE00

    '落下速度ＵＰ（得点 +200点ごとにＵＰ）
    SC01 = Sheets(NM00).Cells(37, 2)
    If SC00 >= SC01 Then
        MX01 = Sheets(NM00).Cells(31, 2)
        If MX01 <= 25 Then
            MX01 = MX01 + 1
            Sheets(NM00).Cells(31, 2) = MX01
            SC01 = SC01 + 200
            Sheets(NM00).Cells(37, 2) = SC01
        End If

    End If
Return

ONTIME34:
    TX01 = Val(Sheets(NM01).Cells(IX02 + 1, IX01))
    '壁の下に空白があれば落下
    If TX01 = 0 Then
        While TX01 = 0
            Sheets(NM00).Cells(IX02 + 1, IX01).Font.ColorIndex = COLOR00(TX00)
            Sheets(NM00).Cells(IX02 + 1, IX01) = Sheets(NM00).Cells(IX02, IX01)
            Sheets(NM00).Cells(IX02, IX01) = ""
            Sheets(NM01).Cells(IX02 + 1, IX01) = Sheets(NM01).Cells(IX02, IX01)
            Sheets(NM01).Cells(IX02, IX01) = ""
            IX02 = IX02 + 1
```

★東京って人が多いですね。朝の千代田線のラッシュにはうんざり！（東京都葛飾区・**初男18歳**）……【**つぐ美**：たしかに千代田線は混んでるわよね。**影**：思い出すな〜。受験で東京に出てきたとき千代田線使ってさ，もうゲンナリで試験もメタメタ。】

```
            TX01 = Val(Sheets(NM01).Cells(IX02 + 1, IX01))
        Wend
        FG00 = 0
    End If
Return

'１秒後に自分自身を再実行
ONTIME40:
    TEXT0 = Now + TimeValue("00:00:01")
    Application.OnTime TEXT0, "ONTIME00"
Return

'終了処理
ONTIME50:
    On Error Resume Next
    For IX00 = 0 To 255
        Application.OnKey Chr(IX00)
    Next

    Application.OnKey "{UP}"
    Application.OnKey "{DOWN}"
    Application.OnKey "{RIGHT}"
    Application.OnKey "{LEFT}"
    Application.OnKey "{ESC}"

    RC00 = MsgBox("GAME OVER", , "END")
Return

End Sub
'NEXT ジュエル予約処理
Sub NEXT00()
    NM00 = "VIEW"
    Randomize Timer

    'ＮＥＸＴ情報保存
     TX00 = Sheets(NM00).Cells(32, 1)
     TX01 = Sheets(NM00).Cells(32, 2)
     TX02 = Sheets(NM00).Cells(32, 3)

    'ジュエルＮＥＸＴ情報作成
    Sheets(NM00).Cells(33, 1) = Int(Rnd() * 6) + 1
    Sheets(NM00).Cells(33, 2) = Int(Rnd() * 6) + 1
    Sheets(NM00).Cells(33, 3) = Int(Rnd() * 6) + 1

    'ジュエル描画処理（パラメータ３が １ ）
    FG00 = TET000(13, 3, 1)

    'ジュエル情報移動
    Sheets(NM00).Cells(32, 1) = Sheets(NM00).Cells(33, 1)
    Sheets(NM00).Cells(32, 2) = Sheets(NM00).Cells(33, 2)
    Sheets(NM00).Cells(32, 3) = Sheets(NM00).Cells(33, 3)

    Sheets(NM00).Cells(33, 1) = TX00
    Sheets(NM00).Cells(33, 2) = TX01
    Sheets(NM00).Cells(33, 3) = TX02

End Sub
'強制終了
Sub STOP00()
    NM00 = "VIEW"
    Sheets(NM00).Cells(36, 1) = 1
End Sub
'並びチェック
Sub LNCHK00(XX00, YY00, XX01, YY01, XX02, YY02)
    NM00 = "VIEW"
    NM01 = "HIDE"

    TX00 = Abs(Val(Sheets(NM01).Cells(YY00, XX00)))

    If XX00 + XX01 > 0 Or YY00 + YY01 > 0 Then
        TX01 = Abs(Val(Sheets(NM01).Cells(YY00 + YY01, XX00 + XX01)))
    Else
        TX01 = 0
    End If

    If XX00 + XX02 > 0 Or YY00 + YY02 > 0 Then
        TX02 = Abs(Val(Sheets(NM01).Cells(YY00 + YY02, XX00 + XX02)))
    Else
        TX02 = 0
    End If

    If (TX00 = TX01) And (TX00 = TX02) Then
        Sheets(NM01).Cells(YY00, XX00) = Abs(Sheets(NM01).Cells(YY00, XX00)) * (-1)
        Sheets(NM00).Cells(YY00, XX00).Font.ColorIndex = 15
        Sheets(NM00).Cells(YY00, XX00) = "※"

        Sheets(NM01).Cells(YY00 + YY01, XX00 + XX01) = Abs(Sheets(NM01).Cells(YY00 +
YY01, XX00 + XX01)) * (-1)
        Sheets(NM00).Cells(YY00 + YY01, XX00 + XX01).Font.ColorIndex = 15
        Sheets(NM00).Cells(YY00 + YY01, XX00 + XX01) = "※"

        Sheets(NM01).Cells(YY00 + YY02, XX00 + XX02) = Abs(Sheets(NM01).Cells(YY00 +
YY02, XX00 + XX02)) * (-1)
        Sheets(NM00).Cells(YY00 + YY02, XX00 + XX02).Font.ColorIndex = 15
        Sheets(NM00).Cells(YY00 + YY02, XX00 + XX02) = "※"
    End If
End Sub
'得点表示
Sub SCORE00()
    NM00 = "VIEW"
```

★私，部長さんなんですよね。何人入部するのか心配ですよ。（福井県武生市・**つかち～。17歳**）……**【編**：何の部長さん？**つぐ美**：パソコン関係のクラブでしょ。**影**：いやいや，華道部？それとも陸上部？かもよ。いずれにしても，自分で部長“さん”はないよな。**】**

```
    SC00 = Sheets(NM00).Cells(37, 1)
    TX00 = "SCORE : " & Trim(Str(SC00))
    Sheets(NM00).Cells(10, 11) = TX00
    Sheets(NM00).Cells(10, 11).Select
End Sub
'色番号設定
Function COLOR00(CC00)
    Select Case CC00
        Case 0
            COLOR00 = 1
        Case 1
            COLOR00 = 3
        Case 2
            COLOR00 = 6
        Case 3
            COLOR00 = 5
        Case 4
            COLOR00 = 46
        Case 5
            COLOR00 = 10
        Case 6
            COLOR00 = 13
        Case 7
            COLOR00 = 4
    End Select
End Function
'キャラクタ設定
Function CHAR00(CC00)
    Select Case CC00
        Case 1
            CHAR00 = "●"
        Case 2
            CHAR00 = "■"
        Case 3
            CHAR00 = "◆"
        Case 4
            CHAR00 = "▲"
        Case 5
            CHAR00 = "▼"
        Case 6
            CHAR00 = "★"
    End Select
End Function
```

《リスト３》Module3.basのプログラム・リスト

```
'ジュエル右移動
Sub RIGHT00()
    NM00 = "VIEW"
```

```
    'ジュエル情報取得
    XX00 = Sheets(NM00).Cells(34, 1)
    YY00 = Sheets(NM00).Cells(35, 1)

    'ジュエル衝突判定（パラメータ３が ２ ）
    FG00 = TET000(XX00 + 1, YY00, 2)
    If FG00 = 0 Then
        'ジュエル消去処理（パラメータ３が ０ ）
        FG00 = TET000(XX00, YY00, 0)
        'ジュエル描画処理（パラメータ３が １ ）
        FG00 = TET000(XX00 + 1, YY00, 1)
        Sheets(NM00).Cells(34, 1) = XX00 + 1
        Sheets(NM00).Cells(35, 1) = YY00
        Sheets(NM00).Cells(30, 1) = 1
    End If
End Sub
'ジュエル左移動
Sub LEFT00()
    NM00 = "VIEW"

    'ジュエル情報取得
    XX00 = Sheets(NM00).Cells(34, 1)
    YY00 = Sheets(NM00).Cells(35, 1)

    'ジュエル衝突判定（パラメータ３が ２ ）
    FG00 = TET000(XX00 - 1, YY00, 2)
    If FG00 = 0 Then
        'ジュエル消去処理（パラメータ３が ０ ）
        FG00 = TET000(XX00, YY00, 0)
        'ジュエル描画処理（パラメータ３が １ ）
        FG00 = TET000(XX00 - 1, YY00, 1)
        Sheets(NM00).Cells(34, 1) = XX00 - 1
        Sheets(NM00).Cells(35, 1) = YY00
        Sheets(NM00).Cells(30, 1) = 1
    End If
End Sub
'ジュエル回転
Sub UP00()
    NM00 = "VIEW"

    'ジュエル情報取得
    XX00 = Sheets(NM00).Cells(34, 1)
    YY00 = Sheets(NM00).Cells(35, 1)

    'ジュエルの並び情報をローテイトする
    ZZ00 = Sheets(NM00).Cells(33, 1)
    Sheets(NM00).Cells(33, 1) = Sheets(NM00).Cells(33, 2)
    Sheets(NM00).Cells(33, 2) = Sheets(NM00).Cells(33, 3)
    Sheets(NM00).Cells(33, 3) = ZZ00
```


★PIC-BASICに期待！（東京都江東区・**ロボット部のマイルド男17歳**）……【**編**：今月から「PICでBASIC」講座も開始されましたよ。よろしくね。**影**：AKI-PIC877モジュールは秋月電子通商で購入可能です。ベーマガからブームを巻き起こしちゃいましょうか。】

```
    'ジュエル描画処理（パラメータ６が １ ）
    FG00 = TET000(XX00, YY00, 1)
End Sub
'ジュエル落下
Sub DOWN00()
    NM00 = "VIEW"

    Sheets(NM00).Cells(31, 1) = 25
End Sub
```

《リスト４》Module4.basのプログラム・リスト

```
'ジュエル衝突判定・描画・壁化処理
Function TET000(XX00, YY00, FC00)
    NM00 = "VIEW" '主要シート名
    NM01 = "HIDE" '主要シート名
    FG00 = 0
    TX00 = Sheets(NM00).Cells(33, 1)
    TX01 = Sheets(NM00).Cells(33, 2)
    TX02 = Sheets(NM00).Cells(33, 3)

    'ファンクションによる処理の振り分け
    Select Case FC00

        'ジュエル消去
        Case 0
            GoSub TET0100

        'ジュエル描画
        Case 1
            GoSub TET0200

        'ジュエル衝突判定
        Case 2
            GoSub TET0300

        'ジュエル壁化
        Case 3
            GoSub TET0400

        'ジュエルクラッシュ
        Case 4
            GoSub TET0500
    End Select

    TET000 = FG00
Exit Function

'ジュエル消去
TET0100:
    Sheets(NM00).Cells(YY00, XX00) = ""
    Sheets(NM00).Cells(YY00 + 1, XX00) = ""
    Sheets(NM00).Cells(YY00 + 2, XX00) = ""
Return

'ジュエル描画
TET0200:
    Sheets(NM00).Cells(YY00, XX00).Font.ColorIndex = COLOR00(TX00)
    Sheets(NM00).Cells(YY00, XX00) = CHAR00(TX00)
    Sheets(NM00).Cells(YY00 + 1, XX00).Font.ColorIndex = COLOR00(TX01)
    Sheets(NM00).Cells(YY00 + 1, XX00) = CHAR00(TX01)
    Sheets(NM00).Cells(YY00 + 2, XX00).Font.ColorIndex = COLOR00(TX02)
    Sheets(NM00).Cells(YY00 + 2, XX00) = CHAR00(TX02)
    Sheets(NM00).Cells(34, 1) = XX00
    Sheets(NM00).Cells(35, 1) = YY00
Return

'ジュエル衝突判定
TET0300:
    FG00 = Val(Sheets(NM01).Cells(YY00, XX00))
    If FG00 = 0 Then
        FG00 = Val(Sheets(NM01).Cells(YY00 + 1, XX00))
        If FG00 = 0 Then
            FG00 = Val(Sheets(NM01).Cells(YY00 + 2, XX00))
        End If
    End If
Return

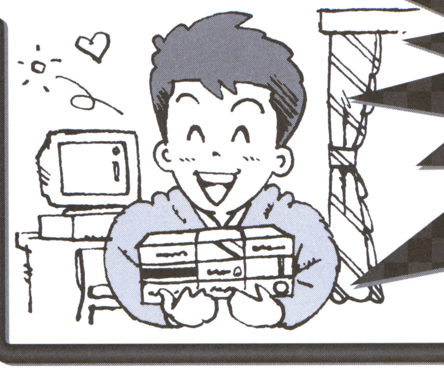
'ジュエル壁化
TET0400:
    Sheets(NM01).Cells(YY00, XX00) = TX00
    Sheets(NM01).Cells(YY00 + 1, XX00) = TX01
    Sheets(NM01).Cells(YY00 + 2, XX00) = TX02
Return

'ジュエルクラッシュ
TET0500:
    Sheets(NM00).Cells(YY00, XX00).Font.ColorIndex = 15
    Sheets(NM00).Cells(YY00, XX00) = "※"
    Sheets(NM00).Cells(YY00 + 1, XX00).Font.ColorIndex = 15
    Sheets(NM00).Cells(YY00 + 1, XX00) = "※"
    Sheets(NM00).Cells(YY00 + 2, XX00).Font.ColorIndex = 15
    Sheets(NM00).Cells(YY00 + 2, XX00) = "※"
Return

End Function
```

★今度は二浪です。去年は仙台の予備校に通っていましたが，今年は地元。来年こそは「やるで！しかし！」来年はN88-BASICを復活させたいです。（青森県弘前市・**スカタン＝ケタロー＝大童19歳**）……**【影：ベーマガに夢中になるのもいいけど，勉強もしっかりな。】**

Windows(F-BASIC V6.0)用



1本で9倍おいしい

ミニゲーム9

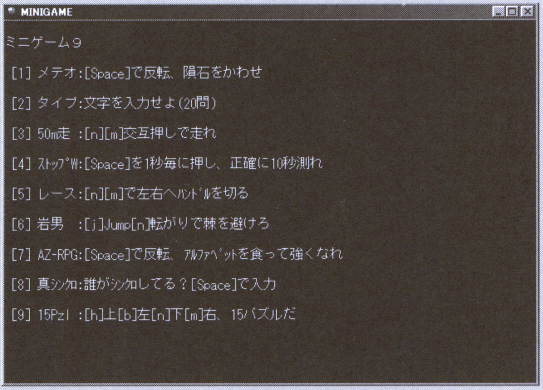
MONOMON(32歳)

はじまり

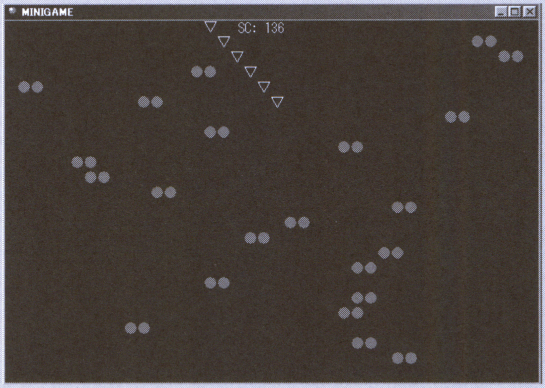
ショック！　F-BASICでもany型宣言でプロシージャの定義ができたのか！

ショック！　GetTickCountよりtimeGetTimeのほうが正確だったのか！(やっとAPIバイブルのマルチメディア編を買った)。

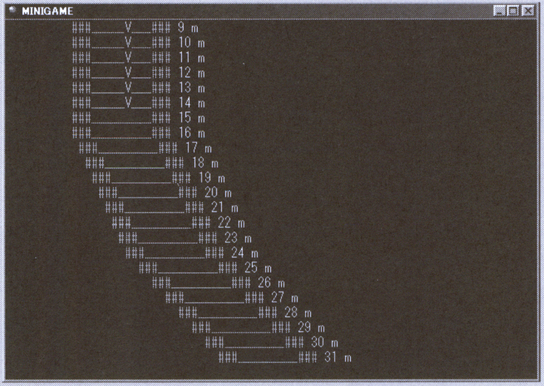
今までの作品を修正してみたら処理が速くなった！　あっ，このアルゴリズム効率悪いぞ！うーん，ダメだ！いったん



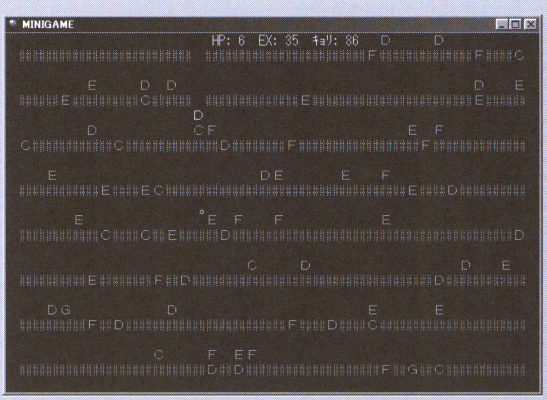
《写真1》九つのミニゲームが用意されている



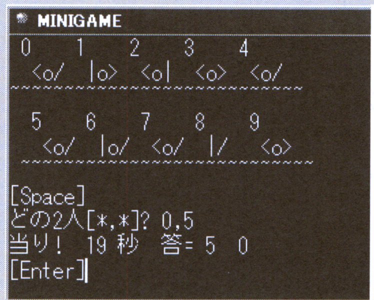
《写真2》懐かしさのある定番のゲームから…



《写真3》テキストで描かれたレースゲームまである



《写真4》これが，作者オススメのゲーム「AZ-RPG」



《写真５》シンクロといった個性的なゲームもオススメ？

原点に帰ろう！

…という事でショートプログラム集です。しかし，そのまま作っても面白くないので，究極まで縮めてみました。どのゲームも１行または２行で完結しています。

入力の仕方

一つの大きなプログラムになってますが，あくまでショートプログラム集です。遊びたいゲームの部分だけ入力して(１～２行)，頭の行番号を0に修正(または最後のgoto 0をgoto 1など行番号に揃える)すれば，それだけで遊べます。

気力のある人はすべて打ち込んでもらえれば，九つのゲームをメニューで選択して遊べます。

注意：F-BASICでは１行255文字(改行コードを含めて257Byte)までとなっています。ギリギリまで詰め込んだので，誌面では改行位置がわかりにくくなると思い，長い行の後に以下の２行を挿入しています。

'<改行>
'

ですから，プログラムの記述してある行は，この目印が出るまでは連続して１行に入力してください。

また，if文など改行する部分を限定してしまう命令は一切使っていないので，命令の区切りであるコロン[:]の部分でさえあれば，改行したり逆に行同士をつなげても問題なく動作します(セミコロン[;]や文字列[""]の中のコロン[:]は違うので注意してください)。

遊び方

タイトル画面で数字キーを押して遊びたいゲームを選択してください(**写真１**)。操作方法は，このタイトルに出ているのでしっかり見ておいてください。

また，ゲームによっては，すぐにスタートするのでご注意ください。各ゲームが終了すると，[Enter]キーでタイトルに戻ります。

なお，注意として，[CapsLock]は解除しておいてください。

各ゲームの説明

[1]メテオ(１行)

ただの隕石避けです。スペースキーで移動方向が反転します。先へ進むとだんだん速くなります。(**写真２**)

評価：×(ひねりがない，１行でもゲームができると判断した試作ゲーム)。

[2]タイプ(１行)

キータイプゲームです。表示されたアルファベットのキーを，すばやく押してください。20問終了すると,時間とミス回数が表示されます。

評価：○(まぁ普通，でも思ったより短いので)。

[3]50m走(１行)

連打ゲームです。[n]，[m] キーを交互に押すと進んでいきます。右端まで進むとゴールし，タイムが表示されます。

評価：△(こんなもんか…)。

[4]ストップW(１行)

スペースキーを１秒間隔で押していって正確に10秒を測るゲームです。

評価：×(基本的すぎてコメントも何も…)。

[5]レース(２行)

最初にサーキット(同じコースを周回する)と公道(ランダムコース)を選択したらスタートします。[n]キーで左，[m]キーで右へ進行方向を傾けます。(**写真３**)

評価：○(ゲームとしては気に入ってるが，ちょっと長い)。

[6]岩男(２行)

前方から迫り来る棘を避けながら進んで行きます。[j]キーでジャンプ，[n] キーで転がります。

評価：△(ジャンプアクションのつもりだったが，ただのタイミングゲームになってしまった。その割には長いかな？)。

[7]AZ-RPG(２行)

《第１表》変数表

今回，ゲームごとに変数が異なるのですが，一応共通で
意味をもたせている変数があるので,それだけ触れておきます。

X	X座標
D	X速度
P	得点
T	時間
M	ミス回数
Q	入力用(「タイプ」では問題用)
H	HP
E	EX

これら以外は基本的にAから順に定義して，
ループ関係はIから順に定義しています。
(絶対にその用途にしか使わないわけではありません)

★5年前まで毎月買っていました。本屋で見かけたのでしばらくぶりに買ってみましたが，ずいぶん変わったような…。私をパソコンとプログラムの世界に導いてくれたベーマガ。いつまでもがんばって！（東京都大田区・**みそ26歳**）……**【つぐ美：毎月買ってね！】**

アルファベットを食べていき，AからZまで成長するゲームです。スペースキーで進行方向を反転します。自分と同じか，それ以上のアルファベットを食べるとEXが1増加，EXが10溜まるたびにレベルアップします。そして，自分より大きいアルファベットを食べるとHPが減少し，小さい場合はHPが増加(回復)します。HPが0以下になるか，Zまで成長するとゲームオーバーです。**(写真4)**

評価：◎(自分なりの解釈でこの手のゲームを作ってみたけど，HP制が思ったより面白かった)。

[8]真シンクロ(2行)

段数を入力すると始まります。段数×5人の人がシンクロの練習をしています。しかし，彼らの中で同じ動きをしている人が2人います。よく見て彼らを発見したらスペースキーで入力モードに入ります。[2,8]など[,]で区切って2人を指定してください。**(写真5)**

評価：○(アイデアはいいけど，妙に淡々としてるので熱中できない。数人で早当てを競うと燃えるかも)。

[9]15Pzl(2行)

普通の15パズルです。[h]キー：上，[n]キー：下，[b]キー：左，[m]キー：右，という操作でパネルをスライドします。左上から右方向へ順番に並ぶように数字を並べ変えればクリアです。

評価：△(アイデアは×，でも3行だったのを2行にまで縮めたので評価アップ。特にシャッフルルーチンは気に入ってます)。

解析の手引き

今回短くするためにいろいろな手を使っています。その中でもわかりにくいものについて触れておきます。

まず[?]ですが，これはprint文の省略形です(マニュアルにも載っています)。そしてprint文の連結記号[;]は省略できるみたいです(マニュアルに載ってない？)。つまり[?A＄B"得点"P]は[print A＄;B;"得点";P]を表しています。

他は，論理式を利用したりandやorを*,+で代用したり，[while inkey＄=""]を[while""=inkey＄]と書いてスペース1文字を減らしたりしています。

一見，式が読みづらい部分も多いですが，以上の点を踏まえれば読み取れると思います。2行プログラムは苦労した処理も多いので，ぜひ解析してみてください(特にmid＄の使い方がいろいろあって面白いと思います)。

最後に

今回のことで，F-BASICの基本仕様すら，まだ使い切れていないことを実感しました。

また，[;]の省略のようにマニュアル外のテクがありそうにも思えます(逆にマニュアルにあるのに使えない機能もあるようですが…例えばsymbol命令の論理操作指定など)。

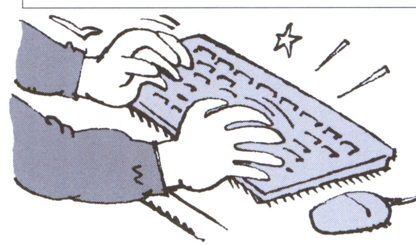
今，一番気になっているのはプログラムガイドのテキスト画面の章に載っている命令で「LINE 文字で線を描く」ってやつです。リファレンスマニュアルにはこんな使い方載ってないよなぁ…誤植かな？

CHECKER FLAG

つぐ美：よく，ここまでミニゲームを制作したものです。また，一つ一つのゲームのリストも1～2行でまとまっているなど，本誌ベーマガらしさがリスト面にも表現されている作品ね。

Dr.D：逆に，短い内容ながらもリストを詰めて記述してある分，解析という面ではチト参考にしにくいリストではある。処理面で触れると，1番の「メテオ」では，PRINT文での改行出力を利用して縦スクロールを行うなど，それぞれの命令の特性をフルに生かしているリストと言える。

影：個人的には，4のストップウォッチゲームが気に入った。このゲームのように1秒ピッタリで止めるほか，9.99秒で止める，といった遊びをした人も多いはず。



F-BASIC

《リスト》ミニゲーム9のプログラム・リスト

```
'0--+----|10--+----|20--+----|30--+----|40--+----|50--+----|60--+----|70--
+----|80--+----|90--+----|100-+----|110-+----|120-+----|130-+----|140-+----
|150-+----|160-+----|170-+----|180-+----|190-+----|200-+----|210-+----|220-
+----|230-+----|240-+----|-End|
'<改行>
'
'#### ﾒﾆｭｰ ####(全2行:249+257=506Byte)
0 cls:A=0:?:?"ミニゲーム９":?:?" [1] メテオ:[Space]で反転、隕石をかわせ":?:?"
[2] タイプ:文字を入力せよ(20問)":?:?" [3] 50m走 :[n][m]交互押しで走れ":?:?"
[4] ｽﾄｯﾌﾟW:[Space]を1秒毎に押し、正確に10秒測れ":?:?" [5] レース:[n][m]で左右
へﾊﾝﾄﾞﾙを切る":?
'<改行>
'
:?" [6] 岩男  :[j]Jump[n]転がりで棘を避けろ":?:?" [7] AZ-RPG:[Space]で反転、ｱﾙ
ﾌｧﾍﾞｯﾄを食って強くなれ":?:?" [8] 真ｼﾝｸﾛ:誰がｼﾝｸﾛしてる？[Space]で入力":?:?"
[9] 15Pzl :[h]上[b]左[n]下[m]右、15パズルだ":while A=0:A=val(inkey$):wend:on A
goto 1,2,3,4,5,6,7,8,9
'<改行>
'
'#### ﾒﾃｵ ####(全1行:242Byte)
1 cls:P=0:X=40:D=2:while screen(X,5)=32:?"▽":for I=1to 15-P¥50:wait
1:next:D=D+2*D*(X=2or X=76or" "=inkey$):X=X+D:P=P+1:locate
int(rnd(1)*37+1)*2,23:color 2:?"●●":locate 35,0:color 7:?"SC:"P:locate
X,5:wend:?"※":input"[Enter]",A$:goto 0
'<改行>
'
'#### ﾀｲﾌﾟ ####(全1行:198Byte)
```

★Dr.J（どくとる・じぇい）は何処へ！？カムバック！！（埼玉県新座市・**Dr.Q17歳**）……【**編**：Dr.Jねぇ。JAVA-WALKと題した「JAVA講座」の講師ですね。2001年6月号から連載開始したんだけど，12月号を最後に尻切れトンボになっている。**Dr.D**：すまん。**編**：えっ？】

```
2 cls:T=0:M=0:for I=1to 20:A=0:Q=int(rnd(1)*26+65):?" "chr$(Q);:while
A<>Q:wait 5:A=asc(ucase$(inkey$)+"."):?chr$(A);:T=T+1:M=M-(A<>46):wend:M=M-
1:?:next:?T/20"秒  ミス:"M:input"[Enter]",A$:goto 0
'<改行>
'

'#### 50m走 ####(全1行:217Byte)
3 cls:A=0:?"ｖｽﾀｰﾄ"spc(70)"ｺﾞｰﾙ
v";:while"n"<>inkey$:wend:?"@";:T=millitime:while csrlin=1:A$="":while(A
and"n"<>A$)+(A=0and"m"<>A$):A$=inkey$:wend:A=1-A:?"@";:wend:T=(millitime-
T)/1000:?T"秒":input"[Enter]",A$:goto 0
'<改行>
'

'#### ｽﾄｯﾌﾟW ####(全1行:257Byte)
4 cls:A$="##.##":?"[Space]":while"
"<>inkey$:wend:A=millitime:M=0:?"<Start!>":for I=1to 10:while"
"<>inkey$:B=(millitime-A)/1000:locate 0:?format$(B,A$)"秒";:wend:T=abs(B-I):?"
誤差="format$(T,A$):M=M+T:next:?using"誤差累計"+A$;M:input"[Enter]",A$:goto 0
'<改行>
'

'#### レース ####(全2行:241Byte+206Byte=447Byte)
5 cls:B$="  ###_________###":C$="656765435675247367 42":?" ｻｰｷｯﾄ[n] 公道
[m]";Q=0:while Q=0:A$=inkey$:Q=(A$="n")-(A$="m"):wend:P=0:X=9:D=0:color 4:for
A=-23to-1:?B$:next:B=0:while screen(X,5)=95:?"V";:for I=1to 15-P¥160:wait
1:next:A$=inkey$
'<改行>
'

:D=D+(A$="n")-(A$="m"):X=X+D:B=B+int(rnd(1)*4-A¥8-B)*(P mod 8=0)*(Q>0):A=A+B-
(asc(mid$(C$,P¥8mod 20+1))-53)*(Q<0):P=P+1:locate A,23:color 4:?B$;:color
7:?P"m":locate X,5:wend:?"*":input"[Enter]",A$:goto 0
'<改行>
'

'#### 岩男 ####(全2行:254Byte+187Byte=441Byte)
6 cls:B$=space$(40):C$=B$:D$="●●◎土丈":P=0:I=0:while I<9:for I=0to 7-
P¥50:for J=0to 9:wait 2:next:A$=inkey$:A=2+(A$="j")-(A$="n"):P=P+1:B=-
(I=0)*int(rnd(1)*5+1)¥2:locate,2:?"  ":for J=1to 2:B$=mid$(B$,3)+mid$("
VVAA",1+(B and J)*2,2):?B$:swap B$,C$
'<改行>
'

:next:color 4:?string$(40,90):color 7:?"SC:"P:color
6:locate,1+A:?kmid$(D$,A,1):?kmid$(D$,A+3,1):color
7:I=I+9*((asc(B$)>32)*(A<3)+(asc(C$)>32)*(A>1)):next:wend:input"[Enter]",A$:go
to 0
'<改行>
'

'#### AZ-RPG ####(全2行:252Byte+251Byte=503Byte)
7 cls:A=0:X=40:P=0:D=2:H=10:E=0:while A<25and H>0:for I=0to 15:wait
1:next:locate X:?"  ":C=(screen(X,6)<>35):D=2*sgn(D+2*D*(inkey$="
")+4*((X=76)-(X=2))):X=X+(C+1)*D:P=P-C:locate 2,22-C:color
2:?string$(76,mid$("  #",1+P mod 3,1)):I=5*C*sgn(P mod 3)
'<改行>
'

:color 4:while I:locate
int(rnd(1)*38+1)*2,22:?knj$(rnd(1)*(P¥20)+P¥30+9025):I=I+1:wend:B=screen(X+1,5
)-96:H=H+(B-A)*(B>-1):E=E-(B>=A):A=E¥10:locate 30,0:color 7:?" HP:"H" EX:"E" ｷ
ｮﾘ:"P:locate X,5:?akcnv$(chr$(A+65));:wend:?:input"[Enter]",A$:goto 0
'<改行>
'

'#### 真ｼﾝｸﾛ ####(全2行:255Byte+253Byte=508Byte)
8 cls:B$="  o><o|<o/|o>|o||o/|> |/ ":C=-1:input"何段[1-6]";Q:Q=abs(Q-1)mod
6:R=5*Q+4:cls:T=time:A=int(rnd(1)*R):B=int(A+rnd(1)*R+1)mod(R+1):while
C<0:D$=mid$(B$,int(rnd(1)*9)*3+1,3)+B$:for I=0to Q:for J=0to 4:K=I*5+J:locate
I+J*5,I*3:?K:locate I+5*J+2
'<改行>
'

:?mid$(D$,1+3*(K<>A)*(K<>B)*int(rnd(1)*9+1),3):next:?tab(I)string$(27,"~"):nex
t:?"[Space]":wait 50:while "=inkey$:input"どの2人[*,*]";C,D:wend:wend:T=time-
T:P=(A=C)*(B=D)+(A=D)*(B=C):?mid$("違う当り",1+4*P,4)"! "T"秒  答
="A;B:input"[Enter]",A$:goto 0
'<改行>
'

'#### 15Pzl ####(全2行:254Byte+257Byte=511Byte)
9 cls:dim A(15):for I=0to 15:A(I)=I:next:B=15:B$="┼──┼──┼──┼──┼":?:?"
[h]上":?" [b]右 [n]下 [m]左":T=-400:while D<16or T<0:locate,4:?B$:D=0:for
I=0to 3:for J=I*4to I*4+3:?mid$(format$(A(J)+1,"┃##┃  "),1-
4*(A(J)=15),4);:D=D-(A(J)=J):next:?"┃"
'<改行>
'

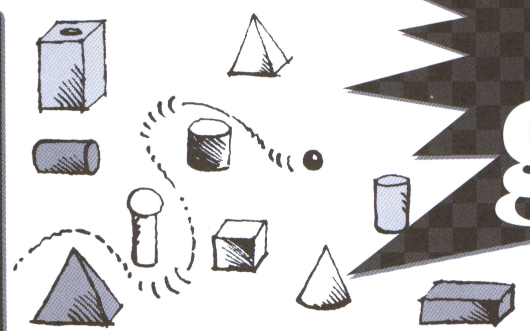
:?B$:next:A$=mid$("nhmb",int(rnd(1)*4+1),-(T<0)):while""=A$and
D<16:A$=inkey$:wend:A=4*((A$="n")-(A$="h"))+(A$="m")-(A$="b"):C=B+A:C=B-A*(C>-
1)*(C<16)*(abs(A)=4or B¥4=C¥4):swap A(B),A(C):T=T-(D<16and B<>C):?T"手
":B=C:wend:erase A:input"[Enter]",A$:goto 0
'<改行>
'

'[EOProgram]
```

★無線LANは快適生活につながる！リビングから引っ張っていたコードのため，ちゃんと閉まらなかった僕の部屋のドアも今はしっかり閉まっている！ウソのよう！（大分県別府市・いしざわん18歳）……【影：しっかり閉まったドア＝快適生活，というのもムナシイ。】

Windows(HSP V2.55)用

HSPでスタンダードゲーム



go zigzag（ゴー ジグザグ）

三好　利治（46歳）

HSP

はじめに

半透明コピーを使ってみたくて，HSP2.55で作ってみました。

左の入口から，障害物を避けながら右の出口へ到達するというもので，どこまでステージをクリア出来るかを楽しむ(?)，とても単純なゲームです。

大昔MSXで作ったものを，記憶を頼りに作りました。

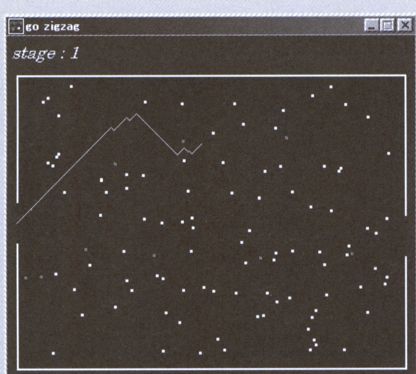
使用方法など

動きは右上45度，右下45度のみです。移動中はスペースキーのみ使用します。スペースキーを押している間は，右下へ移動します。離していると(何もしなければ)右上へ移動します(**写真1**)。

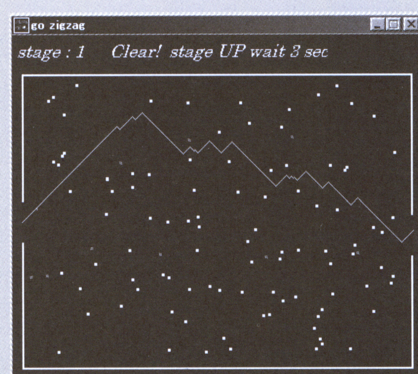
うまくスペースキーを押したり，離したりして障害物を避けて出口へ到達すれば，1ステージクリアです(**写真2**)。ステージの難易度は，ほぼ際限なく高くなります(**写真3**)。

障害物(白いブロック)に当たるとゲームオーバーです(**写真4**)。障害物破壊物(赤いブロック)に当てると，少しブロックが消えます。

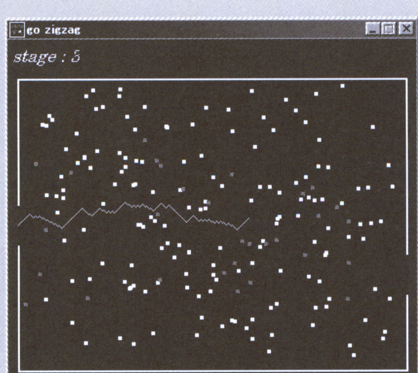
また，ステージアップの度に，出口の位置が変わりますので，最初に注意しておくといいと思います(自分が途中まで気づかなかったもので)。



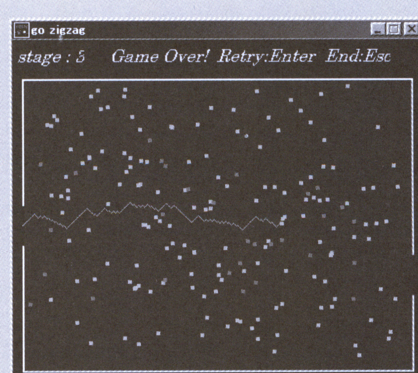
《写真1》定番テキストゲームの一つ



《写真2》画面上のブロックの間をくぐり抜けろ



《写真3》先へ進むごとにステージが難しくなる



《写真4》白いブロックに1度でも当たるとダメ

★最近，ゲームやってません。編集部のみなさんはどうですか？（京都市・**松本昭夫20歳**）……【**影**：もっぱら，投稿ゲームに夢中です。**編**：影さんの場合，仕事なのか，遊びなのか，わからないところがスゴイ！**影**：人は私を鬼のプログラム・チェッカーと呼ぶ。】

入力方法

とても単純でかつ短い**リスト**なので，あっという間に打ち込んで，すぐ遊べます(とりえはそれだけかもしれません)。

バージョン2.5を使われる方は，ソースに書いてあるように，「gmode 3,,,80」をコメントにしてください(ゲームオーバー時の表示以外，まったく同じとなります)。

その他

なんか昔かーしのゲームっぽいし，プログラムの仕組みは解説するほどのものはありませんし，改造などもあまりないのではないかと思います。



CHECKER FLAG

編：定番ゲームの一つですね。正直，内容だけで見れば掲載は厳しい作品なのですが，リストがコンパクトにまとめられている点を考慮しての採用となりました。

影：何回か触れているけど，本誌では作品内容の他，こういったリストが短いといった，リスト面で抜き出ている作品にも注目していくゾ。

つぐ美：赤ブロックを通ると他のブロックが消える，ってアイデアは良かったんだけど，その消えるブロックの数が少ないことや，他の赤ブロックが消されてしまったりなど，このアイデアが，もう一歩生かし切れていない点が残念ね。

Dr.D：ゲーム内で必要な処理ごとに，コメント付きでまとめて記述されているので，プログラム初心者でも解析や改造が行いやすいはずじゃ。試しにmsuやspd，kbxといった変数の値を変更して，ブロックの数や大きさ，移動スピードを変更する改造を行ってみると面白いじゃろう。

《リスト》go zigzagのプログラム・リスト

```
;------------------------------------------
;   go zigzag   HSP2.55版仕様 Toshi.Miyoshi 2002/3
;------------------------------------------
buffer 2,400,300 : color 255,0,0 : boxf
screen 0,420,350,0,dispx-420/2,dispy-350/2
randomize : title "go zigzag" : font "century",18,2

*game_start
color 0,0,128 : boxf
stg=1 : esu=100 : msu=10 : kbx=20 : spd=50 : life=1; 初期値

*stage_up
color 255,255,255 : boxf 10,40,410,340 : pos 5,5 : mes "stage : "+stg
color 0,0,0 : boxf 12,42,408,338 : boxf 10,170,12,210
rnd r,210 : boxf 408,r+50,410,r+90;出口の位置は変化する
;---------- 障害物ブロック 白
color 255,255,255 : ebx=kbx/10;      小数部切捨てのため
repeat esu
        rnd ex,360 : rnd ey,280 : boxf ex+34,ey+44,ex+34+ebx,ey+44+ebx
loop
;---------- 障害物破壊ブロック 赤
color 255,0,0
repeat msu
        rnd ex,340 : rnd ey,180 : boxf ex+14,ey+100,ex+14+ebx,ey+100+ebx
loop
color 255,255,0 : pos 100,5 : mes "Attention! Exit"
x=10 : y=190 : pset x,y : wait 200 : color 0,0,128 : boxf 100,5,400,25
;---------- ゲームループ
repeat
        await spd : stick k,16 : if k=16 : x+ : y+ : else : x+ : y-
        pget x,y : color 255,255,0 : pset x,y;       RGB取得 点を描画
        if (rval=255)&(gval=0)&(bval=0) : color 0,0,0 : boxf x+1,y-15,x+30,y+15 :
continue
        if rval!0 : gosub *game_over : break;        白ブロック又は壁に当った
        if x>=411 : gosub *stg_clear : break;        出口へ到達した
loop

if life=0 : goto *game_start : else : goto *stage_up;ステージアップ又は再トライ

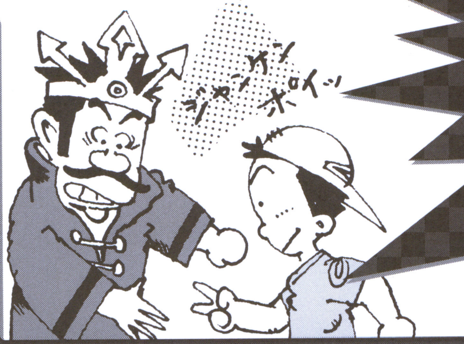
;---------- ステージクリア処理 難易度を上げる
*stg_clear
color 255,255,255 : pos 100,5 : mes "Clear!  stage UP wait 3 sec"
stg+=1 : esu+=30 : msu+=10 : kbx+=5 :if kbx>40:kbx=40 : spd-=5 : if spd<10:spd=10
wait 300 : color 0,0,128 : boxf
return

;---------- ゲームオーバー処理
*game_over
gmode 3,,,80;     ※HSP2.5を使う場合はこの行をコメントにして下さい
                  ;半透明コピーを指定  HSP2.5には実装されていません
gsel 0 : pos 12,42 : gcopy 2,0,0,398,298
color 255,255,0 : pos 100,5 : mes "Game Over!  Retry:Enter  End:Esc"
repeat
        wait 1 : getkey k,13 : if k=1:life=0 : return
          getkey,k,27 :if k=1:end
loop
;---------- //EOF
```


★信号機あるにもかかわらず，自動車の衝突が続発する「魔の交差点」が僕の家の隣に…怖ぁ～！！（福島県郡山市・**近内沖尋19歳**）

……【**つぐ美**：車が家に突っ込んでくるなんてことはないのかな？ほんと，怖そう！**影**：行政は何をやってるんだ！行政は！！】

iモード対応携帯電話(iアプリ)用



じゃんけんを取り入れた格闘ゲーム

i.Doll（アイドール）

三好　和久（21歳）

ストーリー

ここは，iアプリの中の世界。みんな平和に暮らしていました。そこに魔王カズーンが現れた。さあ，やっつけよう！

システム

完全2ROUND先取制です。ターン毎にじゃんけんをし，攻撃側(勝者)，防御側を決定します(**写真1**)。攻撃側は行動値(約10～15)が無くなるまで攻撃ができます(**写真2,3**)。防御側は例外を除き，常に防御しかできません。

操作方法，キャラクター説明などは，別に掲載されている表(**第1表～第3表**)を参考にしてください。

ゲーム開始時のキャラクター選択は，1ボタンで行います。



《写真1》じゃんけんを取り入れた格闘ゲーム



《写真2》奥義や秘技を使って大ダメージを与えろ



《写真3》もちろん，奥義や秘技は，相手も容赦なく使用してくる



《写真4》相手の体力が無くなれば勝ち。まずは2Round先取を目指せ

苦労したこと

容量を10KB以内にする事が一番苦労しました。

実機テストでは，機種ごとに画像表示スピードが異なるので，あまり苦にならない程度に調整するのに苦労しました。

さいごに

格闘ゲームの初心者には少し難しいゲームだと思います。最初は連続で攻撃を入れる練習をしてください。得点増加，評価上昇，ダメージ増加の三つの特典が付いてきます。

CHECKER FLAG

影：iアプリによる格闘ゲームだけど，常にリアルタイムでの対戦でなく，じゃんけんによって攻守を決定しながら対戦する，という点がオリジナリティがあっていいね。

編：どうしても操作レスポンスが遅くなりがちなiアプリですが，この形式なら厳密なコマンド入力などを行わなくても，十分に遊べますね。

Dr.D：リストは，iアプリでありながら超大作ギリギリの,かなりボリュームがある内容。チョット目を通すのも敬遠してしまいそうじゃが，COM動作処理や，リストの頭にある圧縮データからのキャラクター描画処理など，参考になるところが多いリストじゃゾ。

★IE立ち上げたらいきなりMicrosoftが出てびっくり。何だろう?と思い，読むとIE6のダウンロードだった。早速インストール……変わんないなぁ～。(三重県度会郡・Autosuggestion15歳)……【編：見た目は変わんないけど，信頼性などの面で向上してるんだよ。】

《第１表》操作方法

攻撃パート			
ボタン	行動	行動値	説明
1	攻撃	2	通常で使える唯一の攻撃手段です。最大4HIT。
2	奥義	3	LPを半分以上ある時に使用可能。大ダメージを与える
3	秘技	4	HPが赤メータで，LPが半分以上の時に使用可能。各効果は絶大。
4	後退	1	左に１歩移動。相手との距離をとる時に有効
5	前進	1	右に１歩移動。相手に近づく時に有効
6	溜め	1	LPが溜まる。
特殊入力			
5	I.C	２回復	奥義から更に攻撃，奥義，秘技を繋げる事を可能にする仲介技
防御パート			
ボタン	行動	説明	
5	防御	万能防御。成功時は攻撃，奥義，秘技どれでも無効化にする。	
1	カウンター，阿修羅	ユミナの奥義使用回数につき１回使える。相手が攻撃の場合のみ発動。使用回数-1。	
2	カウンター，不動の手	蓮花の秘技使用回数につき１回使える。相手が奥義・秘奥義の場合のみ発動。使用回数-1。	

《第２表》キャラクター説明

名前	奥義	秘奥義	感触
栗丸教師	打撃	打撃(70%ダメージ)	素のままで強いキャラです。
鏡 真人	打撃	打撃	防御力が高めなので，防御練習に役立つ！
レーゼ	打撃	打撃(I.Cで上昇)	攻守共に強い。I.Cからの秘技は絶大。
鍋男	端に逃げる 相手のLP半減	自分の番まで寝る，HPランダム回復 相手のLPを０に	相手の技を封じるのが，このキャラのメイン行動です。 たまに言うことをきかないのが玉に傷。
ユミナ	当て身回数＋1	当て身回数＋1	攻守共に弱い。しかし，攻撃時で当て身回数を稼いだあとの， 防御時でのカウンターが脅威。
シルベン	打撃	ステータス上昇	防御面が△。しかし重複できる秘技は重宝
神代 光	打撃	ステータス上昇	攻守共に△。しかし，奥義のHIT数が多いのでI.Cから攻撃を常に意識。

《第３表》タイミング

繋ぎ内容	受け付け時間	入力タイミングと効果
攻撃→攻撃	標準	相手のHP減少が停止する直前。 攻撃の１発の威力は現在のHIT数に比例上昇する。
攻撃→奥義OR秘技	やや少ない	相手が１歩後退直前。 奥義，秘技に攻撃の威力を上乗せする事ができる。
奥義→I.C(インパクトクロス)	やや多い	相手が膝をついて倒れ，起き上がる直前， さらに連続で技を決める事ができる
I.C→攻撃OR奥義OR秘技	少ない	INPUCT　CROSSと表示した瞬間。
防御	標準	命中する直前

《第４表》変数表

key	現在押されたキーの値
dA[]	攻撃力
t	残り行動値
dD[]	防御力
p	攻撃側かどちらか
dC[]	奥義のHIT数
r	HIT数
dS[]	秘技のHIT数
dd	ボーナスダメージ値
s[]	試合数，ラウンド数，戦闘回数
ct	奥義・秘技メッセージカウント
c[]	使用中のキャラ番号
tt	入ってきたスコア
x[]	プレイヤーとCPUの位置
sc	現在のTOTALスコア
mo[]	現在の動作番号
hs	今までのBESTスコア
v[]	勝利数

《リスト１》Sample.javaのプログラム・リスト

```
import com.nttdocomo.ui.*;
import java.util.Random;
import java.lang.*;
import java.io.*;
import javax.microedition.io.*;

public class Sample extends IApplication{
  public void start(){
    PhoneSystem.setAttribute(PhoneSystem.DEV_BACKLIGHT,PhoneSystem.ATTR_BACKLIGHT_ON);
    MainCanvas gc=new MainCanvas();
    Display.setCurrent(gc);
    new Thread(gc).start();
  }
}
```

★先日組み上げたサークルの自作パソコンが場所を移動させただけで動かなくなった……なぜだろ？（愛知県尾西市・シルク・リープト20歳）……【影：居心地が悪かっただけだろう，きっと。編：影さんがフランス料理屋に行くとおとなしくなるのと一緒か。】

```
class MainCanvas extends Canvas implements Runnable{
  Graphics g;
  Random random=new Random();
  Font font;
  int key,t,p,r,dd=0,ct=0,tt=0;
  long sc=0,hs=0;
  int dA[]={16,14,15,16,13,13,12,25};
  int dD[]={2,6,2,4,1,0,1,4};
  int dC[]={12,6,1,0,0,6,11,30};
  int dS[]={7,13,2,0,0,0,0,0};
  int[][] co=new int[2][2];
  int[] rt=new int[3];
  int[] s=new int[3];
  int[] c=new int[2];
  int[] hp=new int[2];
  int[] lt=new int[2];
  int[] at=new int[2];
  int[] df=new int[2];
  int[] ch=new int[2];
  int[] x=new int[2];
  int[] mo=new int[2];
  int[] v=new int[2];
  //画像の基盤データ
  String ac[]={"0000","0001","0010","0011","0100","0101","0110","0111",
               "1000","1001","1010","1011","1100","1101","1110","1111"};

  //画像データ
  String cg[]={ "aaaeamaeaa","bbdofnaoaa","afaoapaiaa","abbobnacaa","abboanaeaa",
              "aaadacapae","ajbocnacaa","aaajfofkaf",
              "aaambmamaa","dlboajbmaa","aoapaaaaaa","abbobpaaaa","bbboanaaaa",
              "aaacadaoae","caeepigiaa","bofkejaaaa",
              "aaaaafadah","anboblaeae","cadkdmaeae","aaaabcdmdi","abchcgcbbl",
              "acaeaebmak","abaoanadhp","bfdododobm",
              "aaajaoamae","aaaabmbmbm","dohphpejaa","aaabapaoaa","abababadab",
              "aiaidoaiai","dmhihagiea",
              "bmdmhmdebm","inhkboabaa","fafidiaaaa","aaaabmbpae","bbaoanaeaa",
              "acadaoaeaa","cahaccahac","aabebibmca",
              "aiaiaiaiai","bnaoajbmdo","iahagiiedm","aabnboebba","aebfbpbpao",
              "aihicidmca","ajfmaiijck","dgboambgcc",
              "aaefnkfoan","blboajbeai","aaabaoaobc","hoolplolho","bohoaaebcc",
              "bbccccobf","aaeejpemaa","aibmdgbmai",
              "emembobobo","bnaoclbegd","aabadehiam","ababilcdab","ememamboho",
              "caeccdeoce","fpimeajpem","fncccckccfn",
              "aaaibiaiaa","badpfmaoaa","aeapaoaiaa","aabpbmacaa","aabpamaeaa",
              "aaacadaoae","aibpcmacaa","aaaifpfiae"};
  String na[]={"栗丸教師","鏡 真人","レーゼ","鍋男","ユミナ","シルベン","神代 光",
              "カズーン"};
  String nick[]={"火星人","魔重戦士","女レスラー","平成の謎男","合気道",
               "かませ狼","プリンセス"};
  String cMs[]={"1チクチクくるぜ2栗丸アターック3これがぁー4栗丸の拳だぁー5",
              "1地獄に2行っちまいな3太陽神の4怒りを知るがいい5",
              "1超2かかと落し3サンシャイン4マジック5",
              "1ひー、来るなよ!23??について4不て寝する5",
              "1あまいのよ2阿修羅3不動の手よ4かの者を裁け-5",
              "1邪空牙…!2滅3フェンリルパワー4メイクアップ5",
              "1これが正義の光2スターブレイド3我に禁断の魂を!4マインドレスドール5",
              "1輝け、不焔鳥2ロイヤールハント3神の力を知れ4ゴッドブレス5"};
  String sMs[]={"0栗を見るたび1思い出せ2","0貴様には1死んでもらうぞ2",
              "0もうイヤ!1おかしくなりそう2","0ふんっ!1勝手にしろ2",
              "0ぷーっ(>.<)1しつこいわね2","0リベンジは1迷いの森へ2",
              "0正義は勝つ1な~んてね2","0我は理!1頂点の者なり2"};

  public void paint(Graphics g){}

  public void run(){
    long i=0;
    try{
      //iアプリの初期設定
      init();
      for(;;){
        //タイトル表示
        title();setSoftLabel(1,"棄権");
        //キャラ選択
        chSelect();
        do{
          //対戦相手を表示
          nBattle();
          do{
            //ROUND!
            setSoftLabel(0,v[0]+"-"+v[1]);Round();
            //戦闘のメインへ
            main();
          }while(v[0]!=2 && v[1]!=2);
          //どちらかが2勝しました(Winner!)
          g.lock();Rf(7);g.unlock(true);sl(300);
          g.lock();Rc(0);g.fillRect(0,25,120,20);
          ms("Winner!",40,6);g.unlock(true);sl(1000);
          mo[p]=c[p];mo[1-p]=28;x[p]=3+p;x[1-p]=4-p;
          g.lock();Rc(0);cU(0);cU(1);g.unlock(true);sl(1000);
          g.lock();ms("「"+Ha(sMs[c[p]],0),85,0);
          ms(Ha(sMs[c[p]],1)+"!!",100,0);g.unlock(true);sl(2000);
          //1試合ボーナス
          i=(30000+(rt[2]*2500)+Math.abs(random.nextInt()%10000))/(v[0]+v[1]-1);
          if(p==0){ sc+=i;Se("F.BOUNS "+i,1500);Se("TOTAL: "+sc,2500); }
          //倒した相手がラスボスならエンディング!
          if(s[0]==5 && p==0){ Ending();v[1]=2;sc+=100000; }
        }while(v[1]!=2);
        //ゲーム終了時に、今のスコアはハイスコアを越えているかを見る
        if(sc>hs){
          //ハイスコアを越えている(NEW!)
          g.lock();Rf(0);ms("SCORE "+sc,65,6);g.unlock(true);
          Se("NEW RECORD!",3000);
          try{  //スクラッチパッドに書き込む
            DataOutputStream out=Connector.openDataOutputStream("scratchpad:///0");
            out.writeLong(sc);out.close();hs=sc;
          }catch(IOException e){}
        }
        do{
          //ゲームオーバーです。何か押されたらタイトル画面に戻ります
          fkey();g.lock();Rf(0);ms("GAME IS OVER",65,6);g.unlock(true);
        }while(key==0);sl(500);
      }
    }catch(Exception e){}
  }
```


★HSP完全マスターはわかりやすくてよかったです。(大阪市・玉出のタマ36歳)……【**編**:HSP講座,好評ですね。**Dr.D**:講座を開始してから投稿も増えてきているようじゃな。**編**:それもレベルが高い作品ばかり。学校のクラブで使ってるところも多いようです。】

```
//●タイトル画面です
private void title(){
  try{
    //ハイスコアを読み込みます
    DataInputStream in=Connector.openDataInputStream("scratchpad:///0");
    hs=in.readLong();
    in.close();
  }catch(IOException e){}
  //もしハイスコアがなかったらハイスコアを１０００点にします
  if(hs<=0){ hs=1000; }
  do{
    //表示
    g.lock();Rf(4);ms("ｉ．Ｄｏｌｌ",68,7);ms(" 1st. "+hs,115,6);g.unlock(true);
  }while(getKeypadState()==0);
}

//●キャラクター選択画面です
private void chSelect(){
  x[0]=2;s[0]=0;s[1]=0;sc=0;
  do{
    mo[0]=c[0];
    g.lock();Rf(1);ms("どれにする？",40,6);Rc(3);
    g.drawString("(日)   :"+na[c[0]],15,70);Rc(7);cU(0);
    g.unlock(true);Se(""+nick[c[0]],400);
    while(getKeypadState()==0){};
    //キーが押されるとキャラクターが変わって行く
    fkey();if(key==1){ c[0]++; };if(c[0]>6){  c[0]=0; }
  }while(key==1);
}

//●次の対戦相手表示です
private void nBattle(){
  int j;
  int k[]={1,6,2,3,4,0,5,5};
  //戦闘のデータの初期設定
  s[0]++;s[2]=0;lt[0]=20;lt[1]=20;v[0]=0;v[1]=0;rt[2]=0;c[1]=7;
  //対戦相手を決める・・・k[]に依存する
  if(s[0]<5){ c[1]=k[2*s[0]-2+Math.abs(random.nextInt()%2)];  }
  x[1]=6;mo[1]=c[1];
  //アニメーションスタート
  for(j=0;j<=30;j++){ g.lock();Rf(0);
    if(j<=20){  Rc(6);g.drawString("NEXT "+s[0]+"-"+(s[1]+1),20,35);
      if(j%3!=0){
        for(int kk=1;kk<=j%3;kk++){ g.drawString(">",13*kk-13,20);  }
      }
    }
    Rc(7);if(j>10){          cU(1); }
    if(j>20){      ms(na[c[1]],67,7); }
    //最後の戦闘の時にはメッセージを表示
    if(j<11 && s[0]==5){  ms("憐れな生物たちよ…",90,6); }
    if(j>18 && s[0]==5){  ms("罪の報いは死である!",90,6);  }
    g.unlock(true);sl(80);
  }
  //画面がだんだん白くなっていく
  for(j=0;j<254;j+=6){
    g.lock();g.setColor(g.getColorOfRGB(j,j,j));g.fillRect(0,0,120,120);
    Rc(7);cU(1);ms(na[c[1]],67,7);g.unlock(true);
  }
}

//●ＲＯＵＮＤ！の交差表示
private void Round(){
  int i;
  String j[]={"(日)","(月)","(火)"};
  for(i=0;i<=1;i++){  at[i]=dA[c[i]];df[i]=dD[c[i]];ch[i]=dC[c[i]];
    hp[i]=150;co[i][0]=0;co[i][1]=0;x[i]=2+3*i;rt[i]=0;
  }
  s[1]++;s[2]++;t=0;tt=0;
  for(i=0;i<180;i+=3){  g.lock();Rf(12);Rc(7);
    g.drawString("ＲＯＵＮＤ",i-5,70);g.drawString(j[s[2]-1],121-i,70);
    g.unlock(true);
  }
}

//●戦闘のメイン処理です
private void main(){
  int i=0;
  String cp="０Ｃ１Ｂ２Ａ３ＡＡ４ＡＡＡ５Ｓ６ＳＳ７ＳＳＳ８";
  do{
    //両選手ともに立ち画像にする。鍋男が秘技を使っているなら寝る画像にする
    for(i=0;i<=1;i++){  mo[i]=c[i];if(c[i]==3 && co[i][0]==1){  mo[i]=58;  } }
    //ラッシュ数とボーナスダメージを０に！
    r=0;dd=0;uC();
    //行動値が無いならじゃんけんをする
    if(t==0){ janken(); }else{  motion(); }
  }while(hp[1-p]>0);
  //どちらかのＨＰが０を下回ったら↓へ
  v[p]++;sl(50);uC();Sr(9*-(hp[1]),0);
  //そのＲＯＵＮＤの勝負はついた！Ｋ．Ｏ．と表示！！
  g.lock();Rc(0);g.fillRect(0,77,120,20);ms("Ｋ． Ｏ．",88,7);
  //どれだけ余分にダメージを与えたかのＯＶＥＲ ＡＴＴＡＣＫ表示！！
  ms("OVER ATTACK "+(-hp[1-p]),108,1);g.unlock(true);sl(2000);
  //バトル評価計算！！
  i=((rt[0]-800)/rt[1])/150;if(i>7){  i=7;  }
  if(p==0 && i>=0){  Se("評価："+Ha(cp,i),1000);
    if(v[0]==1){  rt[2]=i;  }else{  rt[2]=rt[2]*i;  }
  }
  //最後にそのＲＯＵＮＤでの点数が加算される
  Sr(1500*i,0);
}

//●プレーヤーはボタン待ち、ＣＰＵはＡＩ動作を処理してくれる所
private void motion(){
  tt=0;
  //プレイヤーはボタン待ち
  if(p==0){        fkey();
  }else{
    //ＣＰＵのＡＩ動作決定
    if(x[1]-x[0]==1){
      cpu(0,0,0,50,0,50);if(t>1){  cpu(55,8,8,0,20,9); }
    }else{
      cpu(0,3,2,5,60,30);
      if(lt[1]>24){ cpu(0,8,8,0,75,9);  }
    }
```


★「OfficeでBASICプログラミング」連載が楽しみです。“クリックができれば大丈夫です”と椛島先生が言ってますが本当？クリックならできます。（埼玉県大里郡・**富士雄14歳**）……【**編**：毎月読んでもらえれば，プログラムが組めるようになるよ。よろしく！】

```
  }
  //もし鍋男を使っているなら15%の確率で操作拒否される
  if(c[p]==3 && key>0 && key<4 && t>1 && Math.abs(random.nextInt()%100)<15){
    Se("ふんっ!",400);key=0;t-=2; }
  //移動する
  if((key==4 || key==5) && Bmove(p,key-4)==1){  t--;mo[p]=63+c[p];uC();sl(50); }
  else if(key==6){  Bup();  }   //リミットを溜める
  else if(key==1){  Atc();  }   //攻撃をする
  else if(key==2){  Cho();  }   //奥義を出す
  else if(key==3){  Sen();  }   //秘技を出す
  else if(key==22){ hp[0]=0;v[1]=1;p=1; }  //棄権する
}

//●移動の処理
private int Bmove(int i,int j){   //i:どちらのキャラか j:前進か後退か
  int k=1,zx;
  zx=x[i];x[i]+=(1-2*i)*(2*j-1);
  if((x[i]+1)%9==0 || x[0]==x[1]){  k=0;x[i]=zx;  }
  return k;
}

//●リミットを溜めの処理
private void Bup(){
  for(int i=0;i<=20;i+=2){  g.lock();Rc(2);
    g.drawLine(x[p]*15,70-i,x[p]*15+16,70-i);
    Rc(6);ms("はあああああ!",105,6);g.unlock(true);
  }
  lt[p]+=2;t--;
}

//●攻撃の処理
private void Atc(){
  int i=0;
  if(t>1 && x[1]-x[0]==1){
    //まず相手に防御判定時間を与える
    mo[p]=8+c[p];uC();Bl(16+c[p]);t-=2;
    if(key==1 && co[1-p][0]>0){
      //もし、相手がユミナだったらカウンターが帰ってくる
      co[1-p][0]--;p=1-p;r=1;t=0;Hs(0,at[1-p]*3);
    }else if(key!=5){
      //攻撃が命中した時の処理
      r++;mo[1-p]=25;uC();sl(80);Sr(50*r,0);
      lt[p]+=2;lt[1-p]+=3;mo[1-p]=26;uC();
      //ダメージ計算
      i=at[p]-df[1-p];Dam(i+i*(r-1)/4);dd+=at[p]/3;
      //もし鍋男に殴られたなら相手はパワーアップ
      if(c[p]==3){  at[1-p]++;  }
      //まだHIT数が4未満なら連続で攻撃できる
      if(r<4){
        if(p==0){ Ca();}else{ cpu(10+10*s[0],0,0,0,90-(10*s[0]),0); }
      }
      if(key==1 && t>=2 && hp[1-p]>0){ Atc();
      }else if(hp[1-p]>0){  Bmove(1-p,0);uC();Ca();
        //攻撃後に連続で奥義OR秘奥義を打ち込むか決定する
        if(p==1){ cpu(0,12*s[0],12*s[0],0,100-(24*s[0]),0);          }
        if(key==2){          Cho();     }        //奥義へ
          else if(key==3){  Sen();  }   //秘奥義へ
        }
      }
    }
  }

//●奥義の処理(リミットが半分以上で使用可能です)
private void Cho(){
  if(t>=3 && lt[p]>=25){  g.lock();Rf(15);g.unlock(true);sl(100);
    //特殊系
    mo[p]=29;uC();sl(100);t-=3;lt[p]-=25;
    if(ch[p]==0){ Sr(900,0);
      if(c[p]==3){       //鍋男の奥義
        ct=1;
        do{ Bmove(p,0);mo[p]=34;uC();sl(50);
          mo[p]=42;uC();sl(50);
        }while(x[p]!=0 && x[p]!=7);ct=0;lt[1-p]=lt[1-p]/2;
      }else{             //ユミナの奥義(回数を増やす)
        mo[p]=12;uC();co[p][0]++;
        Se("いくよ![" +co[p][0]+"]",1000);
      }
    }else{  Hs(0,dd+at[p]*3-df[1-p]); }    //打撃系
  }
}

//●秘奥義(HPが赤メーター、リミットが半分以上で使用できる)
private void Sen(){
  int i=0,j=0;
  if(t>=4 && lt[p]>=25 && hp[p]<76){  g.lock();Rf(6);g.unlock(true);sl(100);
    mo[p]=29;uC();sl(100);t-=4;lt[p]-=25;
    if(dS[c[p]]==0){  Sr(1500,0);
      //特殊系
      if(c[p]!=4){  ct=3;mo[p]=47+c[p];uC();sl(1300);ct=4;
        if(c[p]==3){  //鍋男の秘奥義
          lt[1-p]=0;co[p][0]=1;t=0;uC();sl(800);
        }else{
          for(j=0;j<=10;j++){ i=j%2;mo[p]=47+c[p]+8*i;uC();  }
          Se("ステータスUP!",800);
          if(c[p]==5){  at[p]+=5;df[p]+=4;ch[p]+=3;   //シルベンの秘奥義
          }else if(c[p]==6){  at[p]+=4;ch[p]=24;      //神代 光の秘奥義
          }else{  hp[p]=150;  }                        //カズーンの秘奥義
        }
        ct=0;
      }else{  mo[p]=12;uC();co[p][1]++;      //ユミナの秘奥義(回数増える)
        Se("大技よ![" +co[p][1]+"]",1000);
      }
    }else{  Hs(1,dd+at[p]*5-df[1-p]);  }     //打撃系
  }
}

//●奥義、秘奥義の打撃系処理(kで判別,daはダメージ)
private void Hs(int k,int da){
  int dh=0,j=0;
  dh=ch[p];if(k==1){  dh=dS[c[p]];  }
  mo[p]=24;if(c[p]==4){  dh=1;  }
  //相手に向かって走って行く!
  do{ Bmove(p,1);uC(); }while(x[1]-x[0]!=1);
```

★定期切れたし本屋に売っていないので，載っているかどうかすら分かりません。どうしましょう。（広島市・tinari13歳）……**【影：学校の図書館に毎月入れてもらうようお願いしなさい。いいアイデアでしょ。ついでに本屋にも置いてもらって，毎月買ってよな。】**

```
    //相手に防御判定時間を与える
    ct=1+2*k;Bl(31+c[p]+16*k);uC();
    //ユミナのカウンターを食らった時！
    if(key==2 && co[1-p][1]>0){
      co[1-p][1]--;p=1-p;t=0;r=1;Hs(1,da*2);
    }else if(key!=5){ sl(300);
      //命中した！！
      for(j=1;j<=dh;j++){
        //バシバシ！をＨＩＴ数ぶんだけ繰り返す（2/5を越えると画像変更）
        if(j>dh*4/10){  mo[p]=39+c[p]+16*k;ct=2+2*k;          }
        uC();mo[1-p]=25;r++;uC();mo[1-p]=26;uC();
      }
      lt[p]+=3+k;lt[1-p]+=5+2*k;Sr(900+600*k,0);
      //スペンサなら７割ダメージに！（最低ダメージは３０）
      if(c[p]==0 && k==1){  da=(int)(hp[1-p]*7/10);
        if(da<30){  da=30;  }
      }
      Dam(da);ct=0;
      if(k==0){      mo[1-p]=27;uC();sl(200);
        //インパクトクロスの処理
        if(p==0){ Ca();}else{ cpu(100-(12*s[0]),0,0,0,12*s[0],0); }
        if(key==5){ mo[p]=24;lt[p]+=2;t+=2;Sr(150,0);uC();
          //ＩＮＰＡＣＴ　ＣＲＯＳＳ成功！！
          Se("INPACT CROSS!",100);
          if(p==0){ Ca(); }else{  cpu(25,25,25,0,25,0); }
          if(key==1){ Atc();  }          //攻撃をつなげる
          else if(key==2){  Cho(); }     //さらに奥義をつなげる
          else if(key==3){  if(c[p]==2){  dd+=50;  }  //レーゼならダメージアップ
                Sen();}                  //秘奥義をつなげる
        }
      }else{  Bmove(1-p,0); }    //秘奥義をくらわせた後には１歩下がる
    }
  }

  //●キャンセル処理（プレイヤーのキャンセル全般の処理）
  private void Ca(){
    key=0;
    if(getKeypadState()==0){  sl(230);fkey(); }else{  sl(230);  }
  }

  //●防御全般の処理（成功なら５を返す）
  private void Bl(int i){
    key=0;mo[p]=i;
    if(p==1){ Ca();
    }else{
        cpu(0,0,0,100-(10*s[0]),10*s[0],0);
        if(c[1]==4 && key!=5){  cpu(50,50,0,0,0,0); }
    }
    if(r>0){  key=0;  }
    if((key==1 || key==2) && c[1-p]!=4){  key=0;  }
    if(c[1-p]==3 && co[1-p][0]==1){  key=0;  }
    if(key==5){ Sr(150,1);uC();Se("ガード!",0);Rc(3);cU(1-p);sl(300);ct=0;  }
  }

  //●ダメージ処理全般（ＨＰが減る時の処理）
  private void Dam(int j){
    j=j*(6+2*(hp[1-p]/51))/10;
    if(j<4){  j=4;  }
    if(c[p]==2 || c[p]==6 || c[p]==7){
      //神代光、レーゼ、カズーンにはクリティカルがあるよ
      if(Math.abs(random.nextInt()%100)<13){ j=16*j/10;Se("クリティカル!",0);  }
    }
    for(int i=1;i<=j;i++){  g.lock();hp[1-p]--;uH();g.unlock(true);  }
  }

  //●戦闘画面の画像メイン処理
  private void uC(){
    g.lock();Rf(0);
    for(int i=40;i<=70;i++){
      //背景を書く
      g.setColor(g.getColorOfRGB(i*3,i*3,255));g.drawLine(0,i,120,i);
    }
    for(int i=0;i<=1;i++){
      //２人のキャラを表示
      Rc(9-5*i);cU(i);
    }
    //ＨＰとリミットを表示
    uH();
    //行動値を表示
    if(t>0){  ms(""+t,35,7); }else{ Rc(6);g.drawString("VS",54,35); }
    //ＨＩＴ数を表示
    Rc(6);if(r>1){  ms("HIT "+r+"!",50,6); }
    //奥義、秘奥義のメッセージを表示
    if(ct>0){ ms("『"+Ha(cMs[c[p]],ct)+"!!",105,6);  }
    //名前の表示
    Rc(7+s[0]);g.drawString(na[c[0]],50-font.stringWidth(na[c[0]]),35);
    g.drawString(na[c[1]],68,35);g.fillRect(0,70,120,5);
    //スコアの表示
    Rc(8);g.drawString("SR",54,20);
    if(tt>0){ Rc(7);g.drawString(""+tt,50-font.stringWidth(""+tt),20); }
    Rc(6);g.drawString(""+sc,115-font.stringWidth(""+sc),20);
    Rc(0);for(int i=0;i<=4;i++){  g.drawLine(30*i,70,30*i,74);  }
    g.unlock(true);
  }

  //●スコアの計算（Jが正しい時、プレイヤーにｉポイントの得点が入る）
  private void Sr(int i,int j){ if(p==j){ sc+=i;tt=i;rt[0]+=i;  } }

  //●ＨＰとリミットを描画
  private void uH(){
    int j=0,k=0;
    Rc(0);g.fillRect(5,77,110,10);
    for(int i=0;i<=1;i++){
      j=0;
      if(lt[i]>50){ lt[i]=50; }
      if(hp[i]>75){ j=1;  }
      Rc(7);g.drawRect(4+60*i,77,51,5);Rc(4+2*j);if(hp[i]<0){  Rc(12);  }
      k=(int)(50*(hp[i]-75*j)/75);if(k<0){  k=-k; }
      g.fillRect(55+10*i-k*(1-i),78,k,4);Rc(1+lt[i]/25);
      g.fillRect(55+10*i-lt[i]*(1-i),85,lt[i],2);
    }
  }
```


★http://doramanjyu.hoops.ne.jp/　フリーソフト作ってます。投稿しようかな…。（東京都江東区・**どら饅頭15歳**）……【**Dr.D**：ホームページ見たぞ。たいしたもんじゃ。プログラミング入門なんてのも始めようとしているようだし，がんばれ！投稿も待っとるぞ。】

```
//●キャラクターを描画(i側を描画する)
private void cU(int i){
  String mm;
  for(int j=0;j<=4;j++){
    mm=ac[((int)(cg[mo[i]].charAt(j*2))-97)]+ac[((int)(cg[mo[i]].charAt(j*2+1))-97)];
    for(int k=0;k<=7;k++){
      if(mm.charAt(k)=='1'){  g.fillRect(15*x[i]+3*j-6*j*i+12*i,46+3*k,3,3);}
    }
  }
}

//●配列代わりに変数内に区切られている文字を指定してとってくる
private String Ha(String m,int i){
  return m.substring(m.indexOf(""+i)+1,m.indexOf(""+(i+1)));
}

//●CPUのAI動作を送られてきたデータを元にランダムで決定する
private void cpu(int k0,int k1,int k2,int k3,int k4,int k5){
  int cp[]={k0,k1,k2,k3,k4,k5};
  int i=Math.abs(random.nextInt()%100)+1;
  sl(70);key=0;
  do{
    if(cp[key]<i){  i=i-cp[key];key++;  }
  }while(cp[key]<i);key++;
}

//●じゃんけん処理
private void janken(){
  String hand[]={"¥uE693","¥uE694","¥uE695"};
  String vict[]={"勝ち","負け","あいこ"};
  int i=0,j=0;
  sl(100);Se("じゃんけん!",800);
  do{ p=1;j=Math.abs(random.nextInt()%3);
    do{
      Se("㈯"+hand[0]+" ㈷"+hand[1]+" ㈶"+hand[2],0);fkey();i=key-7;
    }while(i<0 || i>2);
    Se(hand[i]+" VS "+hand[j],650);if(i==j){ p=2; }
    //勝敗の判断
    if((i==0 && j==1) || (i==1 && j==2) || (i==2 && j==0)){ p=0;}
    Se(vict[p]+"!",650);
  }while(p==2);sl(200);t=10+Math.abs(random.nextInt()%6);rt[1]++;
  //もし鍋男が寝ていたならば回復しておきる
  if(c[p]==3 && co[p][0]==1){
    co[p][0]=0;mo[p]=c[p];uC();
    i=Math.abs(random.nextInt()%150)+1;hp[p]+=i;if(hp[p]>150){  hp[p]=150;  }
    Se("ナベ男覚醒!<"+i+">",1500);
  }
}

//●Iアプリの初期設定
private void init() throws Exception{
  g=getGraphics();
  font=Font.getDefaultFont();g.setFont(font);
}
```

```
//●影付きで文字を中央表示
private void ms(String mess,int my,int rc){
  Rc(8);g.drawString(mess,60-font.stringWidth(mess)/2,my+1);
  Rc(rc);g.drawString(mess,60-font.stringWidth(mess)/2,my);
}

//●フレーム付きで文字を中央表示
private void Se(String mess,int wa){
  g.lock();Rc(6);g.drawRect(5,90,110,25);Rc(1);g.fillRect(6,91,109,24);
  ms(mess,107,7);g.unlock(true);sl(wa);
}

//●画面をmy色で塗りつぶす
private void Rf(int my){ Rc(my);g.fillRect(0,0,120,140); }

//●KEY入力の判定
private void fkey(){
  int i=0,j=0;
  key=0;j=getKeypadState();for(i=0;i<=21;i++){  j>>=1;if(j>=1){  key++;  }  }
}

//●WAIT処理
private void sl(long sl){
  try{  Thread.sleep(sl); }catch(Exception e){}
}

//●色の変更
private void Rc(int rc){  g.setColor(g.getColorOfName(rc)); }

//●エンディング
private void Ending(){
  String Z="0あんたは偉い!1-制作 三好和久2-ちょっと協力3案 三好孝久4完5";
  int eP[]={120,220,300,315,500};
  for(int j=0;j<=430;j++){  g.lock();Rf(0);Rc(6);
    for(int i=0;i<=4;i++){
      if(eP[i]-j>0 && eP[i]-j<140){ ms(Ha(Z,i),eP[i]-j,7);  }
    }
    g.unlock(true);sl(30);
  }
  sl(3000);
}
}
```

《リスト2》Sample.jamのプログラム・リスト

```
AppClass = Sample
AppName = Sample
AppSize = 10058
LastModified = Fri, 08 Mar 2002 16:45:32
PackageURL = Sample.jar
SPsize = 8
```


★5月号117ページ，台風の中心付近は無風の晴天だと思いますよ，東先生（笑）。（兵庫県神戸市・**岩崎大輔専務。さん21歳**）……
【編：中心付近は台風の目と言いたいわけね。細かいところをついてくる専務さんだこと。まあ，隅々まで読んでくれてありがとう。**】**

PIECE用



日本のスタンダードゲーム

連珠（れんじゅ）

吉田　護（23歳）

内容

内容は，普通の連珠(五目並べ)です。今回は，PIECEのプログラムを初めて作るので，その練習用として考えました。

PIECEの仕様として，仮想画面のメモリ領域を，1次元の配列としてとっているみたいなので，それを2次元の配列として扱えるようにするライブラリも用意しました。

連珠の細かいルール設定に関しては，練習用のプログラムなので見送りました。

使い方・遊び方

起動すると，「Game Start? A:Start」と表示されますので，Aボタンを押して始めてください(**写真1**)。

プレイヤーが先手(黒)です。先に5個，自分のコマを並べれば勝ちです(**写真2～4**)。コンピュータはコマの打てるところにランダムに打ってきます。

操作方法

Aボタン：コマを打つ，決定・終了動作

Bボタン：コンティニュー

方向キー：カーソル移動

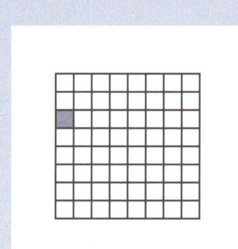
改造のアドバイス・苦労した点

その他のボタンは使用しません。

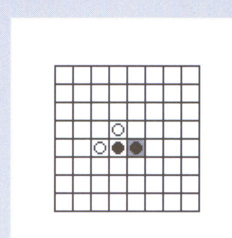
pceAppProcが常に一定周期で実行されるので，ゲームの進行をコントロールするのに，とても苦労しました。

VBのようにイベント駆動のプログラムを記述するプログラムは，かなり楽にプログラムが組めるなと思いました。

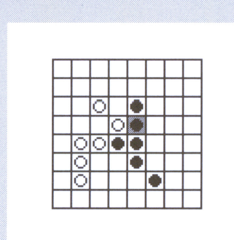
超シンプルな画面構成なので，派手にしてみてください。



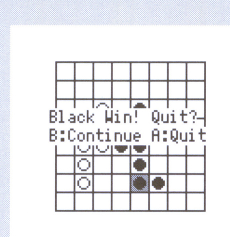
《写真1》この画面で五目並べを行う



《写真2》基本的なゲームなので遊びやすい？



《写真3》COMは弱いので，難しく考える必要がない？



《写真4》PIECEの入門用プログラムとしてはいいかも？

CHECKER FLAG

影：チョット厳しいコメントだけど，ゲームのアイデア，内容，完成度ともにイマイチではある。だけど，PIECE用の単純なゲームプログラム・リストとしては，参考になるところもあるんじゃないかな？

Dr.D:PIECE側で用意されている描画命令の他，Vramの値を書き換えてのグラフィック描画などを行っている点が参考になるのではないかな？また，ゲーム自体が単純なため，PIECEでのゲームプログラムの全体的な組み方という面でも参考にしてほしい。

つぐ美：ただ，練習用プログラムとはいっても，もう少ししっかりした連珠プログラムに仕上げてほしかったわね。はっきり言ってCOM弱すぎ。連珠ゲームになってないわ。

★5月号133ページの下，広告置く前の字が見えてますね。adって書いてありますよ。（大阪市・**広岡聡之16歳**）……【**編**：ごめんなさい。気をつけます。**やすきよ**：読者のみなさん，お久しぶりです。奥村印刷のやすきよです。本当に申し訳ありませんでした。】

《第1表》変数表

OK,NG	TrueかFalseかを示す
PLAYER	プレイヤーのターンとコマの色を示す　※1
COM	コンピュータのターンとコマの色を示す　※1
NON	ゲーム終了とコマが置かれていないことを示す　※1
NOT_BEGIN	ゲーム開始前を示す　※1
SPC_X	五目盤の一番左の座標を示す
SPC_Y	五目盤の一番上の座標を示す
CHECK	コマを置いたマスの前後左右の座標を計算するための定数
CurX	プレイヤーのカーソルのX座標を示す
CurY	プレイヤーのカーソルのY座標を示す
TURN	※1のいずれかが入り，ゲームの状態を管理する
TURN_CNT	コマを打った回数
vbuff	仮想画面用メモリ
chrBan	仮想五目盤用配列

《リスト1》Gomoku.cのプログラム・リスト

```
//五目並べプログラム

#include <stdio.h>
#include <stdlib.h>
#include <piece.h>

//コマ・カーソルの画像
#include "Chips.h"
//描画ライブラリ
#include "byouga.h"

#define OK          1
#define NG          0

#define PLAYER              1
#define COM                 2
#define NON                 0
#define NOT_BEGIN   99

#define SPC_X   24
#define SPC_Y   5

//左上・上・右下
//左  ・    右
//左下・下・右下の座標計算用配列
const char CHECK[8][2] = {
          {-1,-1},{ 0,-1},{ 1,-1},
          {-1, 0},         { 1, 0},
          {-1, 1},{ 0, 1},{ 1, 1}
};

//カーソルの座標
char CurX=0,CurY=0;

//プレイヤーとコンピュータのターン
unsigned char TURN=NOT_BEGIN;

unsigned char TURN_CNT=0;

//仮想画面
unsigned char vbuff[128*88];

//五目盤の仮想配列
char chrBan[8][8]={{0,0,0,0,0,0,0,0},
                   {0,0,0,0,0,0,0,0},
                   {0,0,0,0,0,0,0,0},
                   {0,0,0,0,0,0,0,0},
                   {0,0,0,0,0,0,0,0},
                   {0,0,0,0,0,0,0,0},
                   {0,0,0,0,0,0,0,0},
                   {0,0,0,0,0,0,0,0}};

//五目盤を描画
void DrawBan(void){
          char i,j;

          for(i=0;i<9;i++){
                    //線をVramに引く。(Color,X1,Y1,X2,Y2)
                    pceLCDLine(3,SPC_X,SPC_Y+i*10,80+SPC_X,SPC_Y+i*10); //横線
                    pceLCDLine(3,SPC_X+i*10,SPC_Y,SPC_X+i*10,80+SPC_Y); //縦線
          }
}

//カーソルを表示
void DrawCursor(){
          //盤に置かれているコマにあわせカーソルを描画する。
          switch(chrBan[CurX][CurY]){
          case COM:
                    DrawImage(byCursor[COM],vbuff,CurX*10+SPC_X+1,CurY*10+SPC_Y+1,9,9);
                    break;
          case PLAYER:
                    DrawImage(byCursor[PLAYER],vbuff,CurX*10+SPC_X+1,CurY*10+SPC_Y+1,9,9);
                    break;
          case NON:
                    DrawImage(byCursor[NON],vbuff,CurX*10+SPC_X+1,CurY*10+SPC_Y+1,9,9);
                    break;
          }
}

//コマ描画
void DrawChip(){
          char i,j;
          //盤配列をすべて参照し、コマを描画
          for(i=0;i<8;i++){
              for(j=0;j<8;j++){

DrawImage(byChips[chrBan[i][j]],vbuff,i*10+(SPC_X+1),j*10+(SPC_Y+1),9,9);
              }
          }
```

★久し振りにプログラミングをしようと思い，10年振りくらいに貴誌を購入しました。ゲームソフトの紹介コーナーがなくなっているのが非常に残念ですが，OFコーナーや登場人物は相変わらずで一安心しました。（京都市・**あったん30歳**）……**【影：今後もよろしく！】**

```
}

//コマが置けるかチェック
int CheckSetBan(char x,char y){
	char i;
	char tmpX,tmpY,chkFlg=NG;

	if(TURN_CNT!=0){
		//Aボタンを押したところが、空きならば
		if(chrBan[x][y]==NON){
			//周囲のマスをチェックする。
			for(i=0;i<8;i++){
				tmpX=x+CHECK[i][0];
				tmpY=y+CHECK[i][1];
				//座標がマイナスの値ならば次のチェックへ
				if((tmpX>=0 && tmpX<=7) && (tmpY>=0 && tmpY<=7)){
					//マスが空きでなければループを抜ける。
					if(chrBan[tmpX][tmpY]!=NON){
						chkFlg=OK;
						TURN_CNT++;
						break;
					}
				}
			}
		}
	}else{
		TURN_CNT++;
		chkFlg=OK;
	}

	//一手目は必ずコマを置けるようにする。
	if(chkFlg==OK) return OK;
	else return NG;
}

//勝ちかどうか見る。
int CheckWin(char x,char y,char color){
	char i,tmpX,tmpY;
	int cntRenju=0;			//繋がったコマの数。

	//コマを置いた座標の回りのマスすべてをチェックする。
	for(i=0;i<8;i++){
		cntRenju=1;
		tmpX=x+CHECK[i][0];
		tmpY=y+CHECK[i][1];
		while((tmpX>=0 && tmpX<=7) && (tmpY>=0 && tmpY<=7)){
			if(chrBan[tmpX][tmpY]==color){
				cntRenju++;
			}
			tmpX=tmpX+CHECK[i][0];
			tmpY=tmpY+CHECK[i][1];
		}
		if(cntRenju>=5) break;
	}
	return cntRenju;
}

//ゲームの初期化
void GameInit(void){
	memset(&vbuff,0,128*88);
	for(CurX=0;CurX<8;CurX++){
		for(CurY=0;CurY<8;CurY++){chrBan[CurX][CurY]=0;}
	}
	srand(1000);
	TURN_CNT=0;
	CurX=0;CurY=0;
	//盤の描画
	DrawBan();
}

//勝ち負け表示
void DispKatiMake(int color){
	//勝負が着いているならば結果を表示し、終わりか続けるかを決定
	pceFontSetPos(20,30);
	if(color==PLAYER){
		pceFontPrintf("Black Win! Quit?");
	}else{
		pceFontPrintf("White Win! Quit?");
	}
	pceFontSetPos(20,40);
	pceFontPrintf("B:Continue A:Quit");
}

//////////////////////////////////////////////////////////////////////////////
// PIECEの処理                                                              //
//////////////////////////////////////////////////////////////////////////////
/*プログラムが実行されると最初に呼び出され実行される。
/*初期設定をこの関数内で行う。
*/
void pceAppInit(void){
	memset(&vbuff,0,128*88);
	//仮想画面のメモリ領域設定（仮想画面は「vbuff」という配列にする）
	//ディスプレイ表示ストップ
	pceLCDDispStop();
	pceLCDSetBuffer(vbuff);
	//実際の処理がミリ秒単位で動作させるか設定
	pceAppSetProcPeriod(80);
	//ディスプレイ表示スタート
	pceLCDDispStart();
}

/*実際の処理をこの関数内に記述する。
/*pceAppSetProcPeriod()で設定したミリ秒単位で実行される
*/
void pceAppProc(int cnt){
	long button=pcePadGet();
	char ComX,ComY,RetVal=NG;
```

```
	if(TURN==NOT_BEGIN){
		memset(&vbuff,0,128*88);
		pceFontSetPos(30,30);
		pceFontPrintf("Game Start?");
		pceFontSetPos(33,40);
		pceFontPrintf("A:Start");
		if(button&TRG_A){
			TURN=PLAYER;
			//ゲームの 初期化
			GameInit();
		}
	//プレイヤーのターンであれば。
	}else if(TURN==PLAYER){
		//Aボタンが押下されたら
		if(TRG_A&button){
			//コマが置けるか。
			if(CheckSetBan(CurX,CurY)==OK){
				chrBan[CurX][CurY]=PLAYER;
				TURN=COM;
				//プレイヤーの勝ちか
				if(CheckWin(CurX,CurY,PLAYER)>=5){
					TURN=NON;
				}
			}
		}else{
			switch(button){
			case PAD_LF:CurX--;break;
			case PAD_RI:CurX++;break;
			case PAD_UP:CurY--;break;
			case PAD_DN:CurY++;break;
			}
			if(CurY<0){CurY=7;} if(CurY>7){CurY=0;}
			if(CurX<0){CurX=7;} if(CurX>7){CurX=0;}
		}
		if(TURN!=NON){
			DrawChip();
			DrawCursor();
		}else{
			DrawChip();
			DrawCursor();
			DispKatiMake(PLAYER);
		}
	}else if(TURN==COM){
		while(RetVal==NG){
			ComX=rand()%8; ComY=rand()%8;
			RetVal=CheckSetBan(ComX,ComY);
		}
		chrBan[ComX][ComY]=COM;
		TURN=PLAYER;
		if(CheckWin(ComX,ComY,COM)>=5){
			TURN=NON;
		}
		if(TURN!=NON){
			DrawChip();
			DrawCursor();
		}else{
			DrawChip();
			DrawCursor();
			DispKatiMake(PLAYER);
		}
	}else if(TURN==NON){
		if(button&TRG_A){pceAppReqExit(0);}
		else if(button&TRG_B){
			TURN=NOT_BEGIN;
		}
	}
	pceLCDTrans();
}

void pceAppExit(void){}
```

《リスト2》Byouga.hのプログラム・リスト

```
#include <stdio.h>
#include <piece.h>

/*
	仮想VRAMに配列を描画する。
	引数:*Image	描画するイメージの配列
	     *vRam	仮想VRAM
	     X,Y	描画する位置の座標(左上端)
	     X_Len	描画するイメージの幅
	     Y_Len	描画するイメージの高さ
*/
void DrawImage(unsigned char *Image,
				unsigned char *vRam,
				unsigned char X,unsigned char Y,
				unsigned char X_Len,unsigned char Y_Len){
	char i,j=0;
	for(i=Y;i<Y+Y_Len;i++){
		memcpy(&vRam[128*i+X],&Image[j*X_Len],X_Len);
		//memcpy(&vRam[128*i+X],&Image[j],X_Len);
		j++;
	}
}
```

《リスト3》Chips.hのプログラム・リスト

```
//チップ画像
unsigned char byChips[3][81] = {
	{
		0,0,0,0,0,0,0,0,0,
		0,0,0,0,0,0,0,0,0,
		0,0,0,0,0,0,0,0,0,
		0,0,0,0,0,0,0,0,0,
		0,0,0,0,0,0,0,0,0,
		0,0,0,0,0,0,0,0,0,
		0,0,0,0,0,0,0,0,0,
```

P-ECE

★初めて買いましたが「HP作り」や「CPUの歴史」など，とても勉強になりました。これからも買いつづけようと思います。ただ，プレゼントが寂しいですね。(新潟県上越市・**中原武男19歳**)……**【影：メーカーのみなさん！というわけでプレゼント提供よろしく！】**

```
0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,
},{
0,0,0,0,0,0,0,0,0,
0,0,0,3,3,3,0,0,0,
0,0,3,3,3,3,3,0,0,
0,3,3,3,3,3,3,3,0,
0,3,3,3,3,3,3,3,0,
0,3,3,3,3,3,3,3,0,
0,0,3,3,3,3,3,0,0,
0,0,0,3,3,3,0,0,0,
0,0,0,0,0,0,0,0,0,
},{
0,0,0,0,0,0,0,0,0,
0,0,0,3,3,3,0,0,0,
0,0,3,0,0,0,3,0,0,
0,3,0,0,0,0,0,3,0,
0,3,0,0,0,0,0,3,0,
0,3,0,0,0,0,0,3,0,
0,0,3,0,0,0,3,0,0,
0,0,0,3,3,3,0,0,0,
0,0,0,0,0,0,0,0,0,
}
};

//カーソル画像
unsigned char byCursor[3][81]={
{
1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,
},{
1,1,1,1,1,1,1,1,1,
1,1,1,3,3,3,1,1,1,
1,1,3,3,3,3,3,1,1,
1,3,3,3,3,3,3,3,1,
1,3,3,3,3,3,3,3,1,
1,3,3,3,3,3,3,3,1,
1,1,3,3,3,3,3,1,1,
1,1,1,3,3,3,1,1,1,
1,1,1,1,1,1,1,1,1,
},{
1,1,1,1,1,1,1,1,1,
1,1,1,3,3,3,1,1,1,
1,1,3,1,1,1,3,1,1,
1,3,1,1,1,1,1,3,1,
1,3,1,1,1,1,1,3,1,
1,3,1,1,1,1,1,3,1,
1,1,3,1,1,1,3,1,1,
1,1,1,3,3,3,1,1,1,
1,1,1,1,1,1,1,1,1,
}
};
```

コラム　板橋区立美術館主催ワークショップ

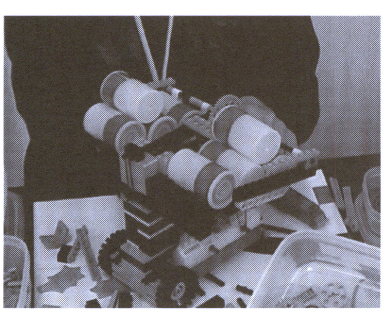
「ロボットをつくろう」

3月30～31日の2日間，東京都板橋区立美術館にてロボット製作のワークショップが行われました。対象は板橋区に在住の小・中学生（10名）。教材は「LEGO MINDSTORMS」のRIS（Robotics Invention System）です。

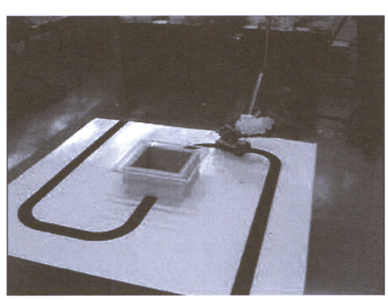
授業は，マインドストームの使い方など，基本的なところから始まり，最終的には自分で競技用ロボットを完成させます。

競技はフィルムケースを運んでから箱に入れ，その数を競うというもので，美術館のオリジナル競技です。ロボットはプログラム搭載の自律型。ライトセンサで床の線を読みとって進み，タッチセンサで箱を検知して投げ込みます。

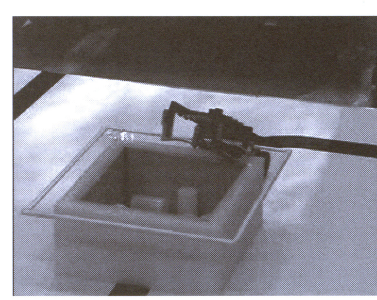
最初は，なかなか難易度の高い競技に思えましたが，中には一度に30個以上ものフィルムケースを入れてしまうような，優秀なロボットが登場して，驚きました。



製作中の1台。積み上げているのがフィルムケース



自動的にフィルムケースを運搬中



見事にシュート成功!!

★小説家を目指しています。推理小説です。書き上げたものを誰かに評価してもらいたいのですが，友達だと何となく恥ずかしいんです。つぐ美さん，お願いできませんか？今度持って行きます。（東京都品川区・**細ちゃん19歳**）……**【つぐ美：**いつでもどうぞ。**】**

★★★★★★★ マイコンBASICマガジン ★★★★★★★

明日のスター・プログラマー

Programmer's Room

今月の投稿をズバリ解析！

編：先月号に引き続き，掲載されているプログラム言語の種類が，非常に多彩な今月の投稿プログラム。さっそく編集部のみんなに，どの作品がベストにふさわしいか，意見を述べていただくとしましょう。

つぐ美：やっぱF-BASICの『ミニゲーム9』がいいわね。リスト面では，まさに本誌ベーマガらしさが出ていますし，ミニゲーム自体も，画面などに派手さはないけど，本質的な面白さはしっかりともっていると思うわ。

影：アイデア面で良かったと思うのは，VBScript+DirectXの『くるくるアイ』かな。3Dフォントを利用した動体視力検査ってのは，なかなか個性的だと思うなぁ。

編：私の意見も述べさせていただきますと，VB+DirectXの『BIKE』は，ゲーム性はもう一歩ですが，3Dレースゲームということで印象に残っていますね。

Dr.D：やはり3Dモノは，見た目にも印象に残りやすいからのぉ。リスト面で注目するとVBの『Action』なんかも，システムに独自性があって良いのじゃが，内容がもう一つ物足りない感があるかな？今月は，ワシも選考に悩んだのじゃが，今回はつぐ美クンが推している『ミニゲーム9』をベストに選ぼうと思う。というのも，最近の投稿プログラムは，どうしてもリストが大きくなりがち。そのアンチテーゼという意味もこめて，こういったワンラインプログラムに近い作品をベストに選んでみた。

編：なるほど。というわけで，今月のベストプログラマーはF-BASIC用『ミニゲーム9』の作者MONOMONさんに決定です。おめでとうございます。

Dr.D：MONOMONさんは，F-BASICの投稿作品だけでなく，ベストプログラマーの常連にもなっているね。これからもプログラム投稿ヨロシク！

●投稿プログラム募集中

自作したプログラムを本誌へ投稿してみませんか？採用されれば，掲載誌＋原稿料(1万円－税金＝9千円)があなたのもの！プログラミング技術の腕試しに，あるいはお小遣いかせぎ(？)に，ぜひベーマガ投稿プログラムコーナーへ参加してみてください。

このコーナーの扉のページ(51ページ)に記載されている応募方法と，以下の事項を参考に，プログラムをeメール，もしくは郵送で応募してください。

★投稿の際の参考事項

・リストの要領は15Kバイトを上限の目安に。それ以上の場合は，超大作として掲載する場合があります。

・作品タイトル名，住所(特に郵便番号)，お名前，年齢は忘れずに記入してください。

・eメール投稿の場合は，プログラムをLHA圧縮して投稿してください。詳しい投稿方法は，ベーマガの公式サイトをご参照ください。ちなみに，メール本文には，何も記入しなくてOKです。

・自分のHPや，VectorなどのWebページで紹介されているプログラムを投稿する場合は，その旨(掲載されているサイト名とURL)を記述してください。また，投稿される作品は，できる限りベーマガ用にアレンジしたものにしてください。

○現在，集中募集中の言語は…

PIECE，HSP(特に3Dを利用した作品)，VBA，互換BASICでの作品を，集中的に募集中。これらのプログラム作品は，掲載率が高いかも？

今月はもう一歩でした

●C++Builder用『High道場』／赤羊旅団(17歳)

Dr.D：このゲームのウリであるバネの動きは，リスト面ではしっかりと作られているのじゃが，ゲーム面から見ると，そのバネの動きが把握しにくいため，イマイチ遊びにくいのが難点じゃな。

●HSP用『メクリ・マクリ』／マンボウ太郎(？歳)

つぐ美：神経衰弱をちょっとアレンジしたゲーム。内容が単純ながらも面白かったのですが，タイマーの設定をプレイヤーがリストを書き換えることで行うってのが難点。この処理をプログラムで自動的に行ってくれるのなら，採用レベルね。

●Visual C++用『ザ・リズム』／堀内克祐(20歳)

影：いわゆる音ゲー形式のゲーム。レベル分けなど，ゲームとしての機能はしっかりしていたけど，この手のゲームでは絶対必要と言えるキー入力のタイミングがイマイチ。もっとダイレクトに反応してくれないと，面白味が半減しちゃうゾ。

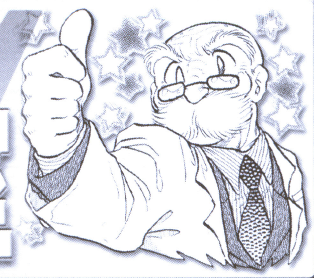
●HSP用『貼り付け君』／はるる(28歳)

編：住所などのテキストデータを保存しておいて，ボタンを押すと，すぐにそのテキストデータが使えるというソフト。実用性はあるのですが，この手のプログラムは，非常に多く存在しているため，やはり何かこのプログラムならでは，というウリがほしいですね。

投稿ありがとう

伊藤　優／南とおれが／井上秀夫／吉田　護／らー抹茶そふと／みゆ／Shota／赤羊旅団／dai256／中田勇／年齢不相応／松本輝彦／M.Hayashi／マンボウ太郎／サバ柄／三好利治／エレフ／はるる／畑の木／shala／MONOMON／橘　一馬／三好和久／金井義男／堀内克祐／
※全員紹介できなくてこめんなさい

Dr.Dのベスト・プログラム研究室



今月のベスト・プログラム：F-BASIC用『ミニゲーム9』（81ページ掲載）

今月のベスト・プログラムは，F-BASIC用『ミニゲーム９』。非常に短いリストで作られたゲームを計９本用意している点と，そのゲームそれぞれが，しっかりとした個性をもっている点が評価された作品。今回は，この９本のゲームの中から，『メテオ』と『AZ-RPG』をピックアップし，それぞれのリスト解説と，簡単な改造方法を紹介していこう。

(Dr.D)

リストを見やすくしよう

このプログラムのいちばんの特徴は，1ラインもしくは2ラインで記述されたリストじゃろう。非常にコンパクトにまとまっているのじゃが，その分リスト内容が把握しにくいのも事実。そこで今回はリスト解説の前にピックアップする２作品のリストを，処理内容はそのままに，見た目にわかりやすいようコメントも加えて記述し直してみた。この後の解説は，掲載してある**リスト１**，**リスト２**に沿って行っていこう。

『メテオ』の解説

この作品のCHECKER FLAGでも少し触れているが，この『メテオ』の特徴である画面の縦スクロールは，PRINT文で出力される改行を利用しておるのじゃ。そのPRINT文は，**リスト１**の行番号10に記述されている。まず，LOCATE命令で，Y座標を画面一番下の23行目へ表示位置を移動させたあと，障害物を表示させるためのPRINT文を実行させることで，画面全体のテキストキャラが，１行分画面上へ移動するというわけじゃ。興味のある人は，試しにこの行のLOCATE文のY座標指定を22へ変更してみよう。画面がまったくスクロールしなくなるはずじゃ(**写真１**参照)。

あと，この『メテオ』の中でわかりにくい処理といえば，行番号8の自キャラの移動量計算の部分じゃろう。これは，内容を解説するよりも**リスト１**にコメントとして掲載している別記述例を見てもらいたい。若干処理内容は異なっておるが，基本的には画面端，もしくはスペースキーが押されたら移動量(変数Dで表される)の正負を変換する，という処理を行えば良いのじゃ。

『AZ-RPG』の解説

このゲームで特徴と言えるのが，落下処理も含めたステージ判定と，各アルファベットとの判定じゃろう。この二つの処理どちらにも共通して使用されている，いわばこのゲームの肝と言える命令が「SCREEN」命令じゃ。この命令は，指定した座標にある文字のキャラクタコードを調べることができる。この命令を使うことで，ステージの床や空白，または各アルファベット判定を行っているのじゃ。では，それぞれの詳しい処理方法を解説していこう。

まずステージ判定は，**リスト２**の行番号7にあるSCREEN命令で行っておる。詳しい処理は，記述してあるコメントを参照してほしい。試しにこのSCREEN文の35という値を変更してみよう。自キャラがステージを示す“#”を無視して画面下へ落下していってしまうはずじゃ。



▲《写真１》『メテオ』のスクロール処理を外したときの実行画面

《リスト1》『メテオ』を記述し直したリスト

```
'#### ﾒﾃｵ ####
1 cls '画面クリア
2 P=0:X=40:D=2 '変数初期化

3 while screen(X,5)=32 '●にぶつかっていない間ループ
4 print "▽" '自キャラの表示

5 for I=1 to 15-(P¥50) '50得点ごとに，スクロールスピードアップ
6 wait 1 'ウェイト処理
7 next I

8 D=D+2＊D＊(X=2 or X=76 or " "=inkey$) '移動量計算
'8行目の処理は，以下のようにも記述できる
'8 if X=2 or X=76 or inkey$=" " then D=D＊-1

9 X=X+D:P=P+1 '自キャラ座標移動＆得点加算
10 locate int(rnd(1)＊37+1)＊2,23:color 2:print "●●" '障害物表示
11 locate 35,0:color 7:print "SC:";P '得点表示
12 locate X,5 '自キャラの表示位置
13 wend

14 print "※" 'ループを抜けたあとの処置＝ゲームオーバー処理
15 input"[Enter]",A$
16 goto 1
```

次は，アルファベットとの当たり判定。これは，行番号20のSCREEN命令を利用して行っておる。この行番号20のSCREEN命令で，自キャラの位置にあるテキストキャラクタの番号を調べる。この時，SCREEN命令に−96という値があるのは，アルファベットAのキャラクタコード96を０と判断するためのものじゃ。つまり，値が０以上であれば，アルファベットと衝突したということになるわけじゃ。

当たり判定が確定したら，SCREEN命令で得た値(Bで示される)と自キャラの状態(Aで示される)を用いて，次の行番号21，22にある命令で，HPとEXの計算を行うわけじゃ。

改造してみよう

このプログラムは非常に短いリストで制作されていることを踏まえて，今回紹介する改造も，非常に手軽に行えるものを紹介していこう。ちなみに，これから紹介する改造方法ならびにリストは，今回掲載している記述し直した**リスト1**，ならびに**リスト2**に沿って行っておる。

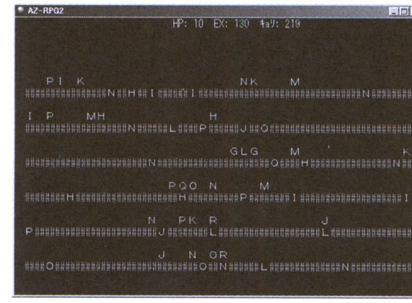
それでは，まず『メテオ』の改造から。ここでの改造は，ウェイト調整，つまりスクロールスピードの調節じゃな。これは『メテオ』に限らず，『AZ-RPG』などでも可能な改造じゃ。方法は**リスト１**の行番号5のfor文にある“15-(P¥50)”の部分を書き換えればOK。15という値を小さくすれば，スピードが上がる。また，50の値を調節すれば，何得点ごとにスピードを上げるか，という部分も調整可能じゃ。

さらに簡単な改造として，行番号10にある障害物表示を行うprint "●●"をprint "●●●●"といったように書き換えれば，好きな数だけ障害物を横に並べることも可能じゃ。

次は，『AZ-RPG』の改造。まずは，ゲームに関連してくるHP，EX計算部部分を改造してみよう。改造リスト例，ならびにその処理内容は，**リスト３**を参照してほしい。

また，ゲームクリアが難しいという人のために，途中からゲームがスタートできる改造も紹介しよう。こちらの改造方法は**リスト４**に掲載してみた。スタート時の実行画面は**写真２**のようになるゾ。

今回は，誌面の都合などもあり，９つすべてのゲームを解説するわけにはいかなかったが，今回紹介したように一度リストを命令ごとに記述し直してみると，ゲーム全体の処理方法が把握できるはずじゃ。他のゲームの処理内容にも興味があるという読者は，ぜひ一度リストの記述直しに挑戦してみてはどうかな？



▲《写真2》『AZ-RPG』でリスト４の改造を行ったときの実行画面

《リスト2》『AZ-RPG』を記述し直したリスト

```
'#### AZ-RPG ####
1 cls:A=0:X=40:P=0:D=2:H=10:E=0 '初期化
2 while A<25 and H>0 'ループ(ゲーム中)
3 for I=0 to 15 'ウェイト
4 wait 1
5 next

6 locate X:print "  "          '自キャラ消去
7 C=(screen(X,6)<>35)          '落下処理用
                               '自キャラの下に#があるC=0，空白C=-1 になる
                               'このCの値を，行番号9の座標計算で使用する

'キー入力による移動処理
8 D=2*sgn(D+2*D*(inkey$=" ")+4*((X=76)-(X=2)))
9 X=X+(C+1)*D '自キャラ座標移動

'ステージ作成
10 P=P-C
11 locate 2,22-C:color 2:print string$(76,mid$("  #",1+P mod 3,1))
12 I=5*C*sgn(P mod 3)

'アルファベット＆空白表示
15 color 4
16 while I
17 locate int(rnd(1)*38+1)*2,22:print knj$(rnd(1)*(P¥20)+P¥30+9025)
18 I=I+1
19 wend

20 B=screen(X+1,5)-96 '自キャラとアルファベットとの当たり判定用
'対応する処理
21 H=H+(B-A)*(B>-1) 'HP増減処理
22 E=E-(B>=A):A=E¥10 'EX増減処理

'ゲーム情報表示
23 locate 30,0:color 7:print " HP:";H;" EX:";E;" ｷｮﾘ:";P
24 locate X,5:print akcnv$(chr$(A+65)); '自キャラ表示
25 wend

'ゲームオーバー処理
26 print:input"[Enter]",A$:goto 1
```

《リスト3》『AZ-RPG』HP,EX増減値の改造リスト例

```
'最後に*2書き加えることで，増減値を２倍にする
21 H=H+((B-A)*(B>-1))*2 'HP増減処理

'¥10を¥2などに変更することで，レベルアップまでの値を変更できる
22 E=E-(B>=A):A=E¥2 'EX増減処理
```

《リスト4》『AZ-RPG』スタート時のパラメータ改造リスト例

```
'Pの値でステージ上のアルファベット表示を，
'Eの値で自キャラの状態を変更できる
'以下の記述だと，ステージのアルファベットはG～Qまでの表示から
'自キャラはNの状態でスタートできる
1 cls:P=200:E=130:A=0:X=40:D=2:H=10: '初期化
```

これから始めるブロードバンド活用術

ADSL HowTo!! 体験記



インターネット電話＆回線チューンナップ

全国各地でサービスが開始されたADSL環境ですが，これと平行して注目のインターネット電話網が急速な広がりを見せています。今回加入した「BBフォン」をもとにして，インターネット電話の利用法を紹介します。さらに，市販ソフトのインストールで，通信速度のアップが実現するADSL回線のチューンナップにもチャレンジしてみました。

音声を分割しインターネット通話 パケット音声通信「VOIP」

前回に引き続きインターネット電話についての紹介からはじめましょう。

インターネットに電話の音声を乗せるには，パソコンのマイクロフォンからの声を，インターネットの通信規格（プロトコル）で，ある単位のデータの固まりに分割し送受信するパケット・データに変換しなければなりません。この音声通信方式を「ボイス・オーバー・インターネット・プロトコル」（Voice Over IP＝VOIP）と呼びます。要するに，音声をウェブ閲覧やデータファイルの転送を行うのと同じデータ転送形式にすることで全世界のインターネット通信網を利用した音声通信（IP電話）が行えるようになったわけです。

VOIPは，インターネット網が最初に普及しはじめた頃には，すでに考案されていた通信方式ですが，高速なデータ通信が行えるADSLが，低価格で利用可能になったことで，最近のような普及のきっかけとなっているわけです。

ADSLでのインターネット電話サービスは，NTTのフレッツ・ADSLをはじめ多くのADSL接続サービス会社が近い将来予定しています。しかし，現在サービスに加入できるのは，Yahoo! BBの

実録 アナログモデム通信から8MbpsADSLへの乗り換えにチャレンジ!! eAccess＋Niftyブロードバンド導入てん末レポートPart③

ADSL回線を引き込んだ部屋はとても狭く，ケーブルを引き回す余裕がなく，導入当初から，モデムとパソコン（ノートPC）をワイヤレス接続，すなわち「無線LAN」でつなぐことを目標にしていました。用意した機器は，イー・アクセスのADSLモデムがルータ機能を持つため，無線アクセス・ポイントを選びました。PCカードも人気の高いアイ・オー・データ機器で統一し，メーカーの違いによる設定のしにくさを極力なくしました。とくに，アクセスポイントは，手のひらサイズの超コンパクト・モデルで，壁掛けにもなり設置場所にも困りません。その導入方法を紹介しましょう。

取材協力・がらん



◀手のひらサイズのコンパクトな無線LANアクセス・ポイント「WN-B11/AXP」（19,800円）を導入環境に合わせ選択



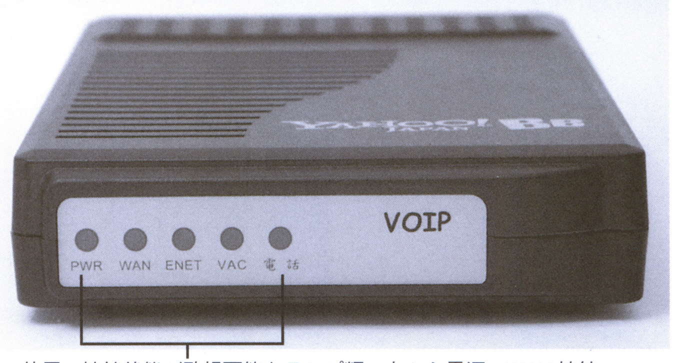
◀インターネット接続は，ノートPCからなので，無線LANアダプタはPCカード・タイプの「WN-B11/PCM」（8,500円）を用意

3月7日 無線LANの導入

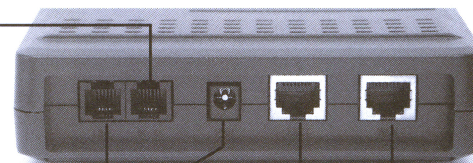
●Yahoo! BBフォン　ターミナル・アダプタの概要



▲Yahoo! BB ADSL接続サービスを導入済みの場合，BBフォンへの加入で，インターネット電話用のターミナル・アダプタが届く。アダプタを追加することで，無料または低料金で通話が行えるようになる



使用・接続状態が監視可能なランプ類。左から電源，WAN接続，イーサネット接続，VAC，電話確認表示



電話端子／これまで使っていた電話機をそのまま接続

ライン端子／ADSLモデムのフォン端子へ接続

電源端子／ADSLモデムと同様のACアダプタ供給

ENET端子／パソコンのネットワーク端子へ接続

WAN端子／ADSLモデムのイーサネット端子へクロスケーブルで接続

「BBフォン」に限られます。

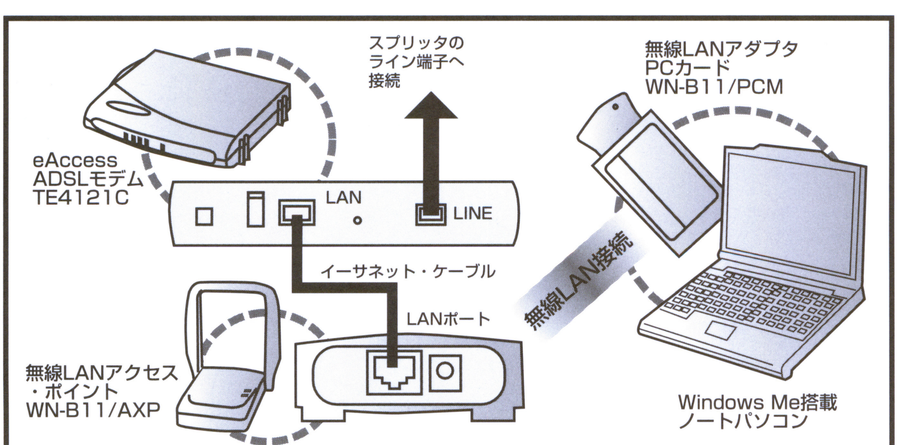
5月号で少し触れていますが，前号の記事締め切り間際に，「BBフォン」への加入がOKになり，数日してインターネット電話用のターミナル・アダプタが到着済みです。送られてきた機材を元に，インターネット電話の導入方法を紹介しましょう。

「BBフォン」導入法と利用の注意ポイント

BBフォンは，単独では利用できません。Yahoo! BBのADSL接続サービスへの加入が前提になりADSLモデムやスプリッタとの接続を行わなくてはなりません。ケーブル類の接続方法については，5月号で掲載した図解を参考にしていただくとして，まずターミナル・アダプタを見てみましょう。

すでに導入済みのADSLモデムと同サイズの外観ですが，前面パネルのランプの内容が多少異なります。「VAC」は，通話中に点灯し，同じBBフォンからのADSL回線を介して呼び出しを受けた場合は，点滅してその旨を知らせます。

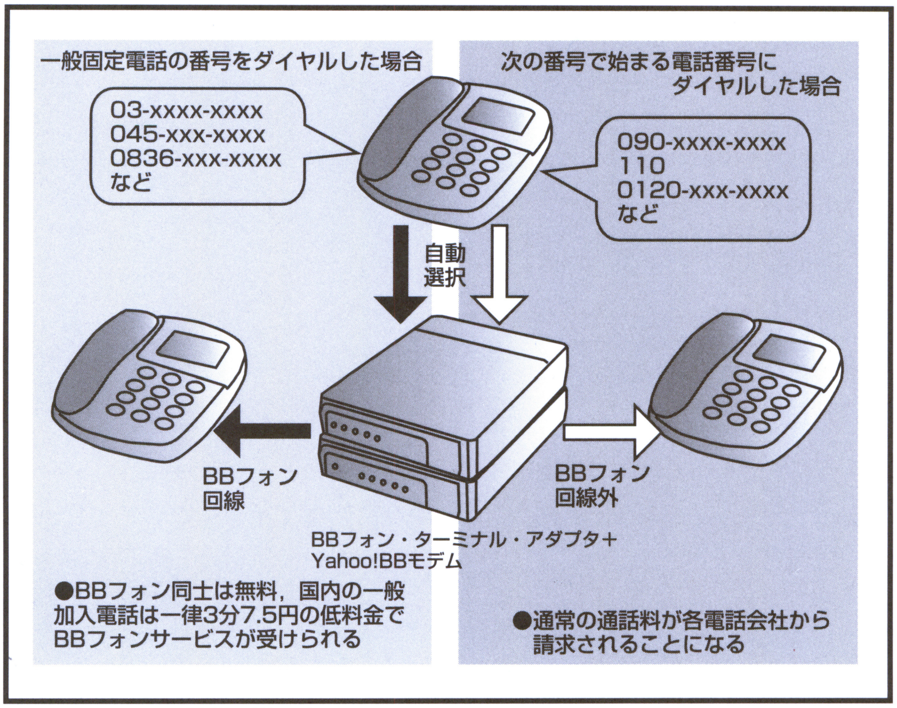
BBフォン以外，すなわちNTTの一般アナログ回線（スプリッタのフォン端子側）からの呼び出し時は，VACランプは消えた状態になります。要するに，相手がNTT回線を利用しているか，BB



▲「@Nifty」＋「イー・アクセス」＋無線LAN環境のシステム解説図



▲イー・アクセスからレンタルのADSLモデムはルータ機能もあり，無線LANアクセス・ポイントをつなぐことで，複数のパソコンから1本の回線がワイヤレスで共有も可能だ



▲BBフォンでの電話の掛け方



▲ADSLモデムとターミナル・アダプタ，電話機の組み合わせでBBフォンを利用する

フォンでADSL回線から呼び出しているかは，このVACランプの状態で確認できるわけです。

VACランプが点滅し，ADSL回線でBBフォン同士で通話する場合は，通話料無料で行えます。こちらから掛ける場合も，VACランプの状態は着信時に準じます。なお，「電話」ランプが消えている場合は，電話を掛けることはできません。

電話機は，これまでスプリッタのフォン端子に接続していたものが，そのまま利用可能で，ターミナル・アダプタ背面のモジュラ端子へ差し替えます。

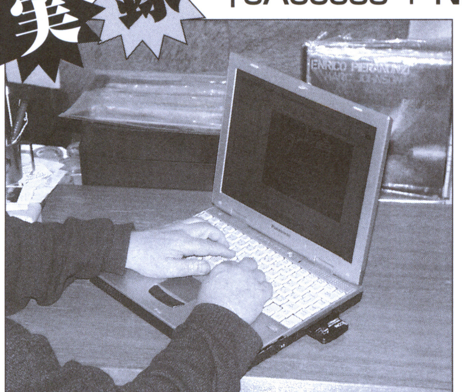
この電話機から相手側の電話番号を選ぶわけですが，通常の電話番号がそのまま使えます。市内同士でも必ず市外局番を付けること以外は，通常の電話と違いはありません。

相手が，BBフォンであるかどうかは，ターミナル・アダプタが調べ，自動的に接続してくれます。また，相手が一般電話の場合でも，相手の最寄りの接続ポイントまでインターネット回線を経由し，そこからNTT回線で接続されることになるので，相手との距離に関係なく全国一律で，3分7.5円の通話料で利用できます。

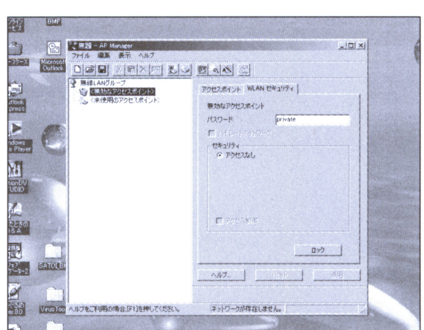
BBフォンでの通話中でも，パソコンからのインターネット接続は，同時に行えるので，1回線で音声通話とインターネット・サーフィンが行える従来のADSL接続とまったく同じ感覚で利用できます。そのうえ，通話料金が通常よりも安くなるので，極めてお得です。

というわけで，早速，これまで使っていたファックス電話機をつないで，インターネット電話を導入したわけですが，

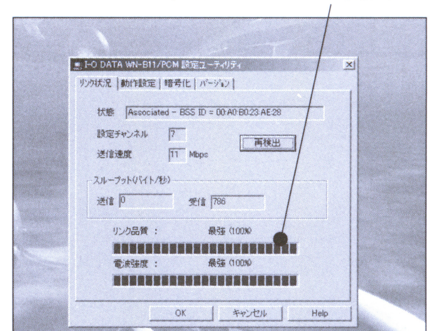
実録「eAccess＋Nifty」ブロードバンド導入てん末レポート③



▲無線LANでのADSL接続は，じゃまなケーブルもなく使い勝手の良さは抜群!!

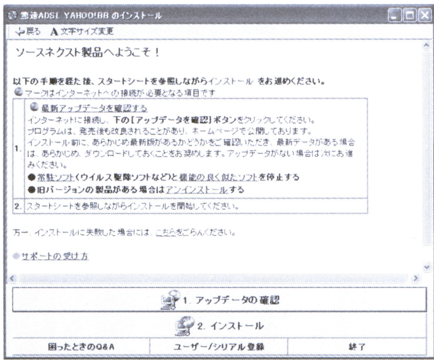


▲無線LANアクセス・ポイント設定用の専用ツール。安全のため不用意なアクセスを防ぐパスワード設定をしておこう

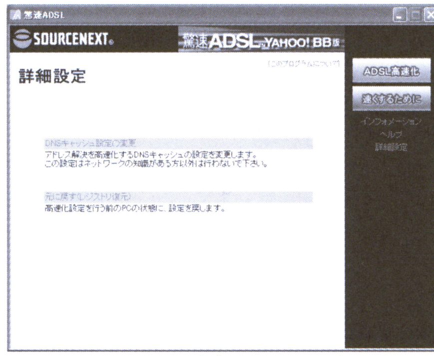


▲無線LANアダプタ（PCカード）設定ツール。電波の受信状況（強度）や使用中の通信速度のチェックが行える

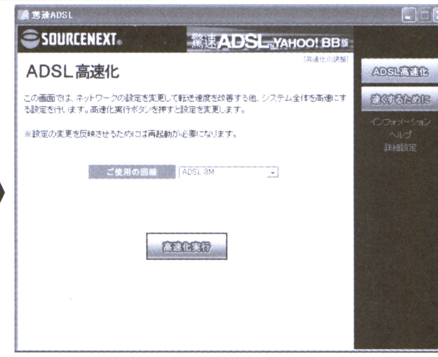
●驚速 ADSL Yahoo! BB 版インストール&ベンチマーク



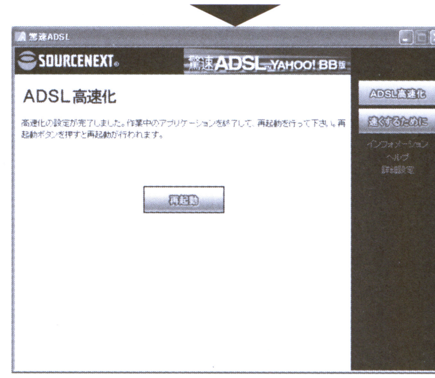
▲導入は，いたって簡単だ。インストーラを起動し「インストール」ボタンを選択



▲DNSキャッシュ設定も可能で，変更したレジストリを復元できる安心機能もある

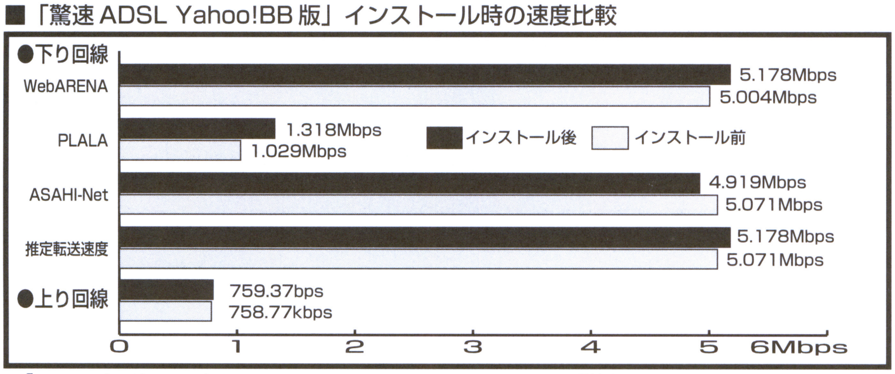


Yahoo! BB 版は，ソフトを起動し「高速化実行」ボタンを選ぶだけ



▲再起動を行うと，RWIN や MUTU などのレジストリを最適値にセット。これで設定完了

■「驚速 ADSL Yahoo!BB 版」インストール時の速度比較



※「BNRスピードテスト」サイト(http://www.musen-lan.com/speed/)にて計測。OSはWindows XP

思いもしないトラブルが…。

なんと，ファックスの送信が出来ないのです。実は，ファックス通信で送受信する信号が，インターネット電話と干渉し，送信が行えないのがその理由のようです。この対策としては，スプリッタのフォン端子を分岐して，ファックス電話を接続するしか手がないようです。このままでは，BB フォンが利用できないので，安価なワンピースフォンを手に入れターミナル・アダプタに接続しました。

ファックス電話を利用する場合は，少々使いづらいようですね。上手い解決法が発見できたら，後日紹介します。

Win95 や 98 の高速化に役立つ加速化ツール・ソフト

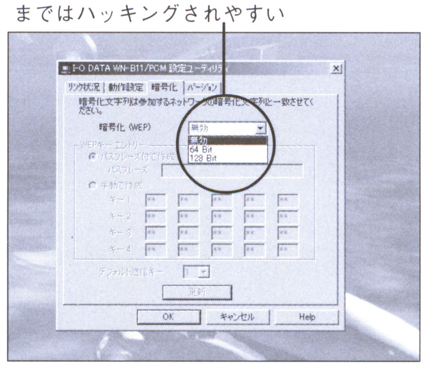
「安くて，速い!!」ADSL 接続サービス

無線LAN接続の各パソコンをアクセス・ポイント側で識別するための SSID。



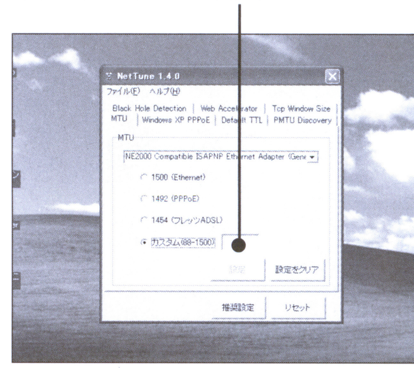
▲アクセスポイントと同一のユーザー ID（SSID）を入力。接続モードは「インフラストラクチャ」を選ぶ

無線 LAN の不正アクセスを防ぐ，通信データの暗号化（WEP）設定。無効のままではハッキングされやすい



▲目に見えない無線 LAN は，暗号化（WEP）設定で万全なセキュリティを行おう。アクセス・ポイントにも同一 WEP を入力すること

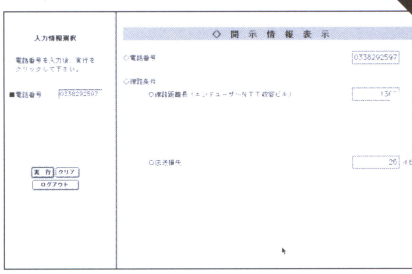
MTU の値を少しずつ変更しながら，最適値を見つけることが基本



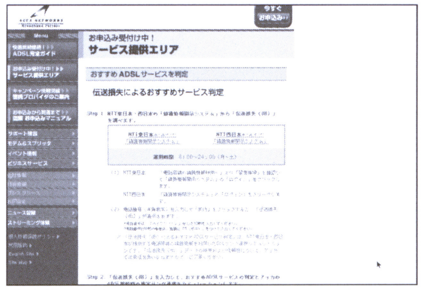
▲RWIN，MTU値調整ツールで高速化を試したが，Windows Me のためか，残念ながら目立った効果は得られなかった

あなたのエリアはどれくらいの通信スピード？ 事前にADSL通信速度がわかる便利な情報サイト

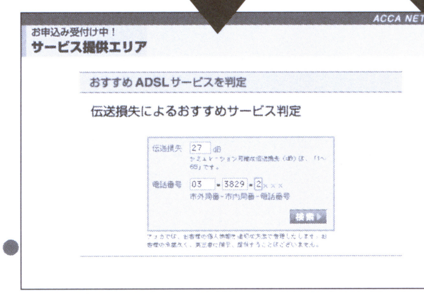
ADSL設備のある最寄りの電話局から2～3km以内が，サービスエリアの目安だが，地図では自宅までの電話線距離は判別しにくい。NTTの「電話回線の路線情報」サイトで事前に確認しておこう。このサイトで得た路線距離と伝送損失の数値で，おおよその通信速度を調べることも可能だ。



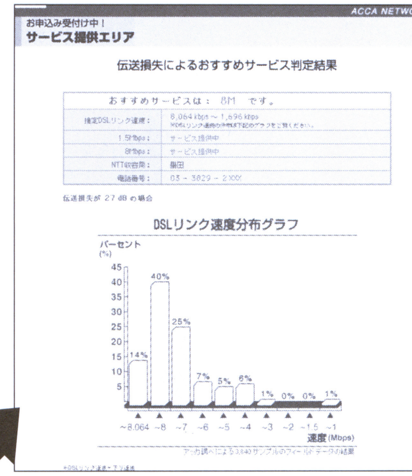
▲NTT「電話回線の路線情報」サイト(http://www.ntt-east.co.jp/line-info/)へアクセスし，電話番号を記入



▲アッカの「サービス提供エリア・サイト(http://www.acca.ne.jp/area/chk/db.html)へ飛び「電話回線の路線情報」サイトで得た伝送損失または路線距離の選択を行う



◀伝送損失で調べる場合は，伝送損失値と電話番号を記入



▲接続速度の予想計算値をバーグラフで表示。おおよそのADSL通信速度が確認できる

ですが，工夫することで，通信速度を速めさらに快適なインターネット接続が楽しめます。このテクニックは，以前に紹介済みなのですが，もう一度紹介しておきましょう。

ADSL回線が持つ通信速度を十分に引き出すには，それに適したパソコンのＯＳ設定を上手く調整することが欠かせないテクニックになります。とくに，Windows 95や98といったブロードバンド環境が整っていなかった時代のＯＳでは，初期設定のままではADSLのスピードを生かし切れません。

スピードアップを図るための設定調整の核になるのが，「RWIN」や「MUTU」値です。これらは，インターネットの接続規格「TCP／IP」のパケット・データを効率良く送受信するための数値で，レジストリというＯＳの設定項目に登録されています。

Windows 95や98は，今となっては通信速度の低速なISDN回線などに合わせてたRWINやMUTU値になっています。高速なADSL回線を利用する場合，そのままの設定値では，本来の性能は出し切れません。そこで，専用のソフトウエア・ツールで最適な数値に調整することになります。使用料無料のオンラインソフトで調整することも可能ですが，専門知識がないと逆に遅くなってしまうこともあります。

安心して導入できるのが，「加速化ツール」と一般に呼ばれる市販のADSL高速化ソフトです。導入しているADSL接続サービス別や，回線の方式ごとにソフトを選べる，さらに簡単な選択を行うだけで自動的に最適な設定を選べるなど，初心者でも安心して高速化が図れるのが特徴です。

今回は，ソースネクスト社から発売の「驚速ADSL」(4,900円）を試してみました。フレッツ・ADSL（NTT東日本／NTT西日本)，アッカ・ネットワークス，イー・アクセス，J-DSL(ODN)の1.5Mbpsまたは8Mbpsに対応し，Yahoo! BB専用の「驚速ADSL Yahoo！BB版」(同価格）もあります。

同じADSL環境なのに，友だちよりも転送速度が劣っていたり，接続サービス会社が保証している速度が得られないなどで，悩んでいる人は試してみてください。なお，インストールしたからといって，急激な速度アップは期待できない場合もあります。また，ブロードバンド回線対応済みのWindows MeやXPでは，効果はそれほどないので，事前にパソコン環境やADSLの種類を調べておきましょう。

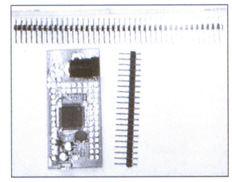
さて，次回は複数のパソコンでADSL回線を共用できるなど，超便利な「ブロードバンド・ルータ」の導入法を紹介します。　**（文・福場龍夫）**

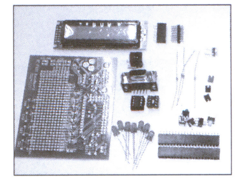


▲回線の共用やファイアウォールなど，使える機能満載のブロードバンド・ルータ

チャレンジ!!
PIC-BASIC

『AKI-PIC877モジュール用ベースボード・キット』（16文字×2行液晶モジュール，シリアルポートx1，タクトスイッチx5，I2C 1MbitEEPROM，LEDx8，AKI-PIC877モジュールx1 付属）4,600円*





『AKI-PIC877モジュール』（8bitマイコン「PIC16F877-20pt」搭載，動作クロック20MHz，A/D機能内蔵，8KBフラッシュROM内蔵，256ByteEEPROM内蔵，RS232CドライバIC，5Vレギュレータ実装済）(単品の価格)1,700円*
『PIC-BASICモジュール』2,700円
*価格はPIC-BASICが付属しないタイプのものです。
*PIC-BASICを利用するためにはBASICインタプリタが書き込まれたAKI-PIC877モジュールが必要です。　（株）秋月電子通商　http://www.akizuki.ne.jp/

ワンチップマイコンでBASICインタプリタが動く

「AKI-PIC877モジュール用ベースボード・キット」

製作・文／松原拓也

「PIC-BASIC」は1チップマイコンにBASICインタプリタを搭載した画期的な開発環境です。四則演算，条件判断，変数の代入（変数はByte/Word/Long型。配列が使えます），I/O制御，A/D入力，LCD表示，EEPROMの読み書きなど，盛り沢山の機能を備えています。

開発方法

(1) パソコンに「PIC-BASIC」の開発ツールをインストールします。対応OSはWindowsです。
(2) ベースボードとパソコンをシリアルケーブル（Dsub9pinストレート）で接続します（写真1）。
(3) スタートメニューから「PIC-BASIC」の開発ツールを起動します（写真2）。
(4)「ツールーオプション」をメニュー選択して，シリアルのポート番号を設定します。
(5) ベースボードのジャンパJ5をプログラムモード（ショート）にします（写真3）。
(6) DCジャックに電源を接続します（2.1 ϕ，7～12V，内側がプラス）。
(7)「エディタ・ウインドウ」にBASICプログラムを記述します。変数や関数名は「ワークスペース・ウインドウ」に表示されます。
(8) ツールバーの「実行」ボタンを押すとプログラムがフラッシュROMに転送・実行されます。「アウトプット・ウインドウ」には，コードのサイズ，エラー情報が表示されます。

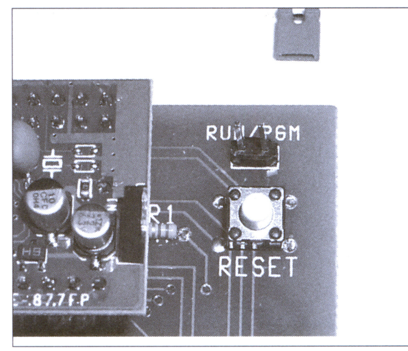
実行中，「一時停止」ボタンを押すと，プログラムが一時停止します。「ウォッチ・ウインドウ」に変数名を登録しておくと，その変数の値を見ることができます。

「ブレークポイントの設定/解除」ボタンで，ブレークポイントを設定しておくと，その実行地点で一時停止します。

プログラム内に「DEBUG」命令を組み込んでおくと，「デバッグ・ウインドウ」に変数や処理の様子が表示されます。

「停止」ボタンでプログラムは停止します。
(9)「最終書き込み」ボタンを押すと，ベースボードとパソコンの接続を切り離せる状態になります。ジャンパJ5をRUNモード（オープン）にしてリセット，または電源を入れ直して実行します。

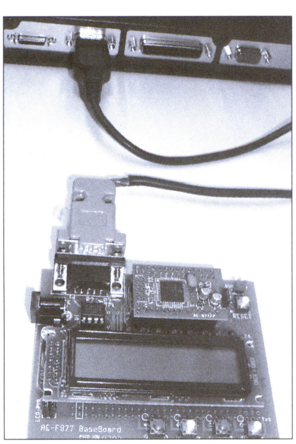
以上でプログラム完了です。

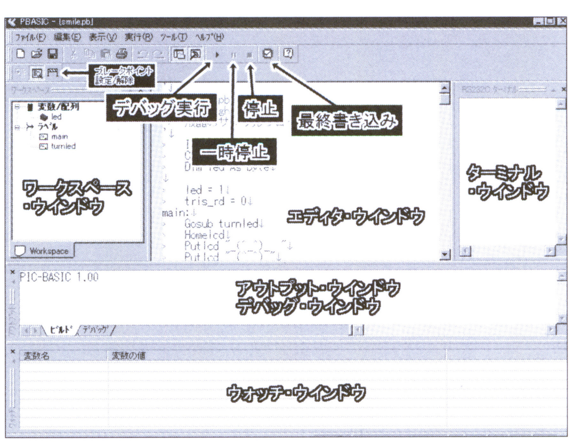


《写真3》ジャンパJ5で実行モード設定

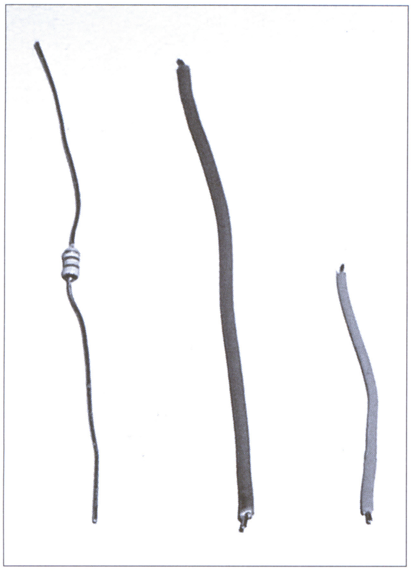
スイッチの増設

ベースボード上にあるスイッチSW2，SW3，SW4は，拡張用のパーツとして


《写真1》シリアルケーブルでベースボードとPCを接続



《写真2》「Visual Basic」風で親しみやすい「PIC-BASIC」用開発ツール



《写真4》必要なパーツ（抵抗とリード線）

付属するものなので，ポートには接続されていません。

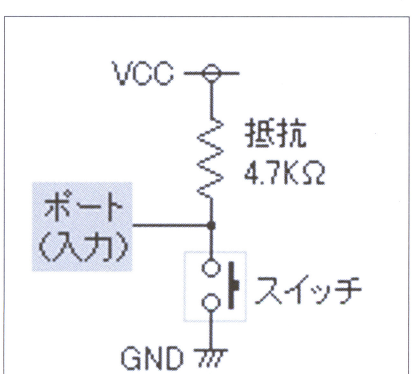
そこで，これらのスイッチを実際に使えるようにしたいと思います。用意するパーツは約4.7kΩの抵抗とリード線（**写真4**）です。工作道具としてハンダと，ハンダコテが必要です。

これから作る回路（**第1図**）の原理について考えてみましょう。

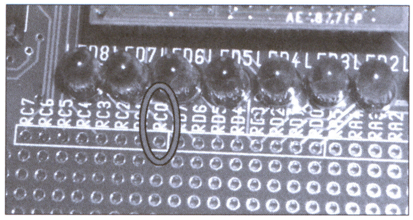
まず，スイッチがオフの場合，電流はほとんど流れません（入力インピーダンスが高い）。そのため抵抗では電圧降下が起こらず，ポートにはVCCの電圧がそのまま入ります。

逆にスイッチをオンにするとどうなるのでしょう？　この場合，グランドに電流が流れるので，ポートにはGNDの電圧＝0Vが入ります。

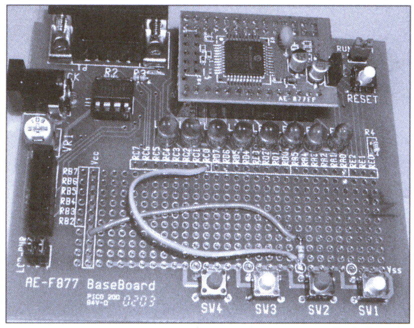
つまりスイッチを押すとLow，離すとHiがポートに入力されます。このようなVCCと抵抗を使った仕組みを「プルアップ」といいます。



《第1図》プルアップの回路図



《写真5》丸の部分がポートRC0



《写真6》スイッチSW2をポートに接続

SW2の取り付け

ベースボードに搭載されているマイコンにはRA，RB，RC，RD，REというポートが備わっています。まず，**第1表**を参考にして，その中から未使用のポートを探しましょう。ポートは空いていればどこでもかまいませんが，今回は「RC0」にしてみました（**写真5**）。

次にベースボードの基板を見てみましょう。表面に「RA*」や「RB*」といったシルク印刷がありますね？　ここがポートの端子です。同様にVCCには「VCC」，GNDには「VSS」と印刷されています。

基板にはユニバーサルエリアという，自由に使える領域がありますので，回路図を参考にして，パーツを取りつけます。ハンダゴテを使って，スイッチと抵抗を接続してください（**写真6**）。

《第1表》ポートの対応表

ポート番号	ポートの割り振り
RA0	（未使用）
RA1	（未使用）
RA2	（未使用）
RA3	（未使用）
RA4	（未使用）
RA5	（未使用）
RA6	（未使用）
RA7	（未使用）
RB0	SW1
RB1	J5
RB2	RS(LCD)
RB3	E(LCD)
RB4	D4(LCD)
RB5	D5(LCD)
RB6	D6(LCD)
RB7	D7(LCD)
RC0	（未使用）
RC1	（未使用）
RC2	（未使用）
RC3	SCL(EEPROM)
RC4	SDA(EEPROM)
RC5	（未使用）
RC6	（未使用）
RC7	（未使用）
RD0	LED1
RD1	LED2
RD2	LED3
RD3	LED4
RD4	LED5
RD5	LED6
RD6	LED7
RD7	LED8
RE0	（未使用）
RE1	（未使用）
RE2	（未使用）
RE3	（未使用）
RE4	（未使用）
RE5	（未使用）
RE6	（未使用）
RE7	（未使用）

スイッチ入力プログラム

リスト1はスイッチSW2を入力するためのサンプルプログラムです。実行するとスイッチの状態を液晶画面に表示します。

ポートには「入力」「出力」という2種類の向きがあります。スイッチの状態を読み込むには「Input」命令を使って「入力」に設定しなくてはいけません。ポートRC0の状態を知るには「rc.Bit0」と記述しま

《リスト1》「スイッチ入力テスト」プログラム・リスト

```
' AKI-PIC877 ベースボード
'
        INITLCD                         ' 液晶モジュールの初期化
        CLEARLCD
        rd = &hff
        INPUT rc.0              'ポートRC0を入力に設定
        WHILE(1)
                SETPOS 0,0
                IF rb.BIT0 = 0 THEN
                        PUTLCD "SW1=on  (RB0=0)"
                ELSE
                        PUTLCD "SW1=off (RB0=1)"
                ENDIF

                SETPOS 0,1
                IF rc.BIT0 = 0 THEN
                        PUTLCD "SW2=on  (RC0=0)"
                ELSE
                        PUTLCD "SW2=off (RC0=1)"
                ENDIF

                SLEEP 100  'ウエイト
        WEND
```

す。たとえば，IF文と組み合わせると，

```
IF rc.Bit0 = 1 THEN
(式)
ENDIF
```

このようになります。値は1ならばHi，0ならLowを意味します。

ゲームに挑戦

早速，スイッチSW1～2を使ったゲームを作ってみました（**リスト2**）。ゲームといっても液晶画面（16文字x2行）は小さく，文字しか表示できませんので，オーソドックスな「旗あげゲーム」にしてみました。モノクロ液晶なので旗は「赤/白」ではなく「右/左」と言い分けてます。

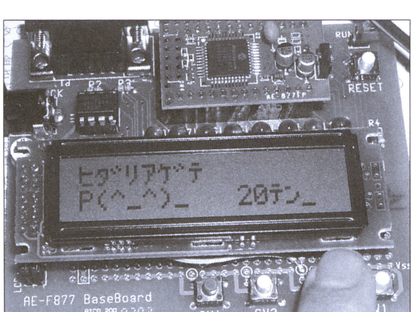
操作方法

ゲームは最初に以下のメッセージが表示されます（5種類）。

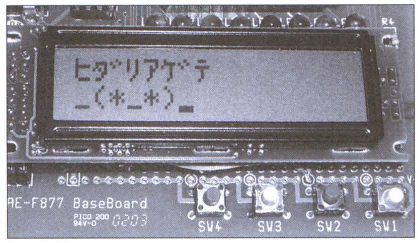
「ﾐｷﾞｱｹﾞﾃ」
「ﾋﾀﾞﾘｱｹﾞﾃ」
「ﾘｮｳﾎｳｱｹﾞﾃ」
「ﾐｷﾞｱｹﾞﾅｲ」
「ﾋﾀﾞﾘｱｹﾞﾅｲ」

そして，一定時間以内にメッセージと同じ旗の状態にすると点数が加算されます（**写真7**）。間違えるとゲームオーバーです（**写真8**）。

SW1を押すと右の旗があがり，SW2を押すと左の旗があがります。両方押すと，両方の旗があがります。



《写真7》メッセージのとおりに旗をあげると得点



《写真8》間違えた旗をあげてしまうとゲームオーバー

_(^_^)P　右の旗をあげた状態
P(^_^)_　左の旗をあげた状態
P(^_^)P　両方の旗をあげた状態
_(^_^)_　何もあげない状態

技術的な解説

座標X,Yに「A」という文字を表示させたい場合，PIC-BASICでは，このようなプログラムになります。

```
SETPOS X,Y
PUTLCD "A"
```

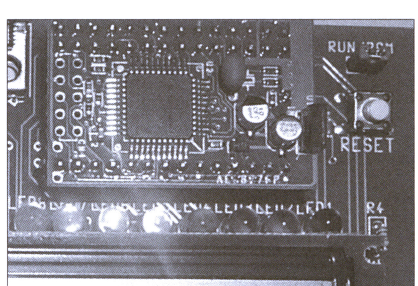
「SETPOS」「PUTLCD」はN88-BASICでいうと「LOCATE」「PRINT」にあたります。

表示させてからの時間調整は「SLEEP命令」を使っています。

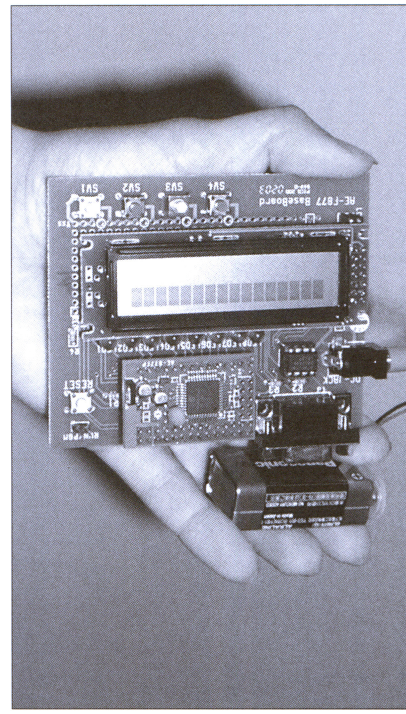
このゲームでは，メッセージが表示されてからスイッチを確定するまでに残り時間を設けています。その残り時間がわかるよう，LEDにバーグラフ表示をさせています（**写真9**）。ポートは1バイトがLED8個（8bit）に割り振られていますので，

```
rd = 0
```

とすることでLED8個が全点灯します。そして，それぞれのビットを1にすると消



《写真9》LEDを使って残り時間を表示



《写真10》9V（6P型）電池だとこんなにコンパクト

灯されます。LEDが順番に消灯していく処理は，シフト命令“<<”を使いました。

あと，困ったことに，PIC-BASICは乱数を発生させるための命令がありません。そこで，「Lookup」命令を使って乱数テーブルを作成してみました。Lookup命令は，指定した(n+1)番目の値を変数に代入することができます。例えば，

```
Lookup
cnt,rand,1,2,3,4,5,6,7,8,9,10,11,12
```

このように記述して，もしcntが0ならば，1番目の値「1」が参照されて，変数randに「1」が代入されます。この数値をデタラメに並べれば，乱数テーブルの代わりになります。しかし，決まったテーブルを循環しているだけでは，すぐに先の値が読めてしまいます。どうしたらいいでしょう？

別の解決方法として，

```
rand = (rand+1) mod 5
```

このような命令をキーが押されている間，呼び続けてみるといいでしょう。ユーザーがキーを押しているタイミングは未知数なので，うまく乱数のように扱えるはずです。

改造

　このままでは，少々ゲームが簡単すぎるようなので，改造してみましょう。方法はいろいろありますが，待ち時間を減らしてみたり，「ﾐｷﾞ ｻｹﾞ ﾅｲ」「ﾋﾀﾞ ﾘｻｹﾞ ﾅｲ」というような「ひっかけ問題」を増やすと，ゲームが難しくなると思います。PIC-BASICはプログラムの修正と実行が簡単に行えますので，実際にプレイしながら改良することができます。

　あと，このAKI-PIC877ベースボードは9V電池でも動作させることができます(**写真10**)。最終書き込み後は，携帯ゲーム機のように持ち歩いてみると楽しいのではないでしょうか。

◆「PIC-BASIC WORLD」
http://picbasic.jp/
　PIC-BASICの開発元（株）エスアイ創房が運営しているPIC-BASICポータルサイトです。

《リスト2》「旗あげゲーム」プログラム・リスト

```
' 旗上げゲーム for AKI-PIC877 ベースボード

        Dim a As Byte
        Dim q As Byte
        Dim cnt As Byte
        Dim time As Int
        Dim rand As Byte
        Dim score As Int

        Initlcd                         ' 液晶モジュールの初期化
        Input rc.0 'ポートRC0を入力に設定
        tris_rd = 0 'ポートRDを出力に設定

        score=0
        rand = 0
MAIN:
        cnt = (cnt + 1) Mod 12
        Lookup cnt,rand,1,0,4,3,4,1,2,0,1,1,3,2 '乱数発生!?

        Clearlcd
        Setpos 0,1
        Putlcd "_(^_^)_"
        Sleep 600

        Setpos 0,0   '出題
        If rand=0 Then
                Putlcd "ﾐｷﾞ ｱｹﾞ ﾃ "
                q=1
        Endif
        If rand=1 Then
                Putlcd "ﾋﾀﾞ ﾘｱｹﾞ ﾃ "
                q=2
        Endif
        If rand=2 Then
                Putlcd "ﾘｮｳﾎｳｱｹﾞ ﾃ"
                q=3
        Endif
        If rand=3 Then
                Putlcd "ﾐｷﾞ ｱｹﾞ ﾅｲ "
                q=0
        Endif
        If rand=4 Then
                Putlcd "ﾋﾀﾞ ﾘｱｹﾞ ﾅｲ"
                q=0
        Endif

        rd = 0  'LED全点灯
        For time=1 To 8
                Sleep 120
                rd = rd << 1
                rd = rd | 1  'LED消灯

                Setpos 0,1
                If rb.Bit0 = 0 Then
                        If rc.Bit0 = 0 Then
                                Putlcd "P(^_^)P"
                                a=3
                        Else
                                Putlcd "_(^_^)P"
                                a=1
                        Endif
                Else
                        If rc.Bit0 = 0 Then
                                Putlcd "P(^_^)_"
                                a=2
                        Else
                                Putlcd "_(^_^)_"
                                a=0
                        Endif
                Endif
        Next

        If q<>a Then
                Setpos 1,1
                Putlcd "(*_*)"
                Sleep 3000  '失敗
                score = 0
        Else
                score = score + 10
        Endif

        Setpos 10,1
        Putlcd score,"ﾃﾝ"

        Sleep 500
        Goto MAIN
```

もっと身近にプログラミング

チャレンジ!!

オリジナル・プログラム言語の制作！

その5 (最終回) 変数システム強化で，配列も，文字列も！

文 まかべひろし

常時接続が一般的になり，ようやくネット時代が本格的になって来ました。ネット技術がプログラマには欠かせないぞ，と言われて数年。ライブラリは整備され，環境も良くなり，当時の苦労はどこへやらという感じで，ネット技術を扱うことができるようになりました。さぁ，いよいよ，ネット専用ソフトが当たり前になるぞ。全国のプログラマよ，立ち上がろう！

絵 桜部のとす

どうも，お疲れさまでした～。2年半も続いてたのですね。そのあいだにアリスもかなり変わりました（＾＾；
頭身伸びたりイロイロと……毎回描いてて楽しかったです。
アリスとネコは，またなんか機会があったら描くと思いますので，どこかで見かけたら，そのときはよろしくですよ～

catlikeホームページ ▶ http://catlike.marchen.to/

ただの数値変数じゃ物足りない，やっぱり多彩な変数が無くっちゃ！ そんなあなたに贈る，配列と文字列変数で，幸せになりましょ～。

熱い季節は近いぞ！

6月号です。雨の時期ではありますが，これが過ぎれば待望の夏です。この，どんよりした雲はどこへやら，突き抜けるような青い空が，まばゆいばかりの太陽が，一面に広がる海が，ボクらを待っているのです！

そんな魅惑的な季節が来ても，プログラマは肌身離さずコンピュータを持っていたいものです。ここは一つ，PDAでのプログラミングなんてどうですか？いえ，今回の話には全く関係ないんですけど，この号が出る頃には，待望のiPAQ PocketPCが出てるなぁって。待ち遠しくてたまらないのですよ。

最後の機能は，なんだっけかな？

えっと，閑話休題，オリジナルプログラム言語の話に戻りますね。前回の記事で，ラベルコンバータまで搭載し，懸案であったGOTO文やGOSUB/RETURNの実装まで行ったベーマガ・マクロプロセッサ（以下BMP）。すでに，数値変数ベースの処理は問題なく行えるくらいの機能はついています。

そろそろ，何を機能追加すれば良いかわからなくなってきた頃。こんなときは，仕様を見直して，次に何を組むべきかを確認してみるのです。

おお，すでに残りはたったの１項目じゃないですか。「文字列変数や配列も欲しいところ」…そう，変数処理に，文字列変数が使えた

ベーマガ・マクロプロセッサ略称「BMP」仕様草案

★整数値を格納できる変数システム
★演算処理は四則演算対応
★条件分岐にIF文。もちろん入れ子対応
★BASICライクにGOTO命令でジャンプ
★サブルーチン処理用にGOSUB/RETURN
・文字列変数や配列も欲しいところ
★デバッグ用に，文字列を画面出力
★マクロ内にコメントを書けるようにする
★は，実装済み

り，配列変数が使えたりしたら便利ですよね。特に配列変数は，テーブル系の処理を行うのに欠かすことができない存在です。

文字列変数はというと，実は，データベース的な処理をしない限りはほとんど使うことはありません。ゲームにおいて文字列変数を用いるというのは，メッセージ周りと，主人公の名前を記憶しておく程度にしか使わないのではないでしょうか。最悪，文字列といえど数値の固まりですので，数値変数だけで処理できないこともありません。ですが，まぁ，文字列変数といえば，ほとんどの処理系にも搭載されているシステムですので，BMPにも搭載しておくに超したことはないでしょう。

配列変数の考え方

一口に配列変数と言いましたが，これはどのように実装すべきなのでしょうか。プログラムを組むときは，コードの書き方よりもまず，実現したい内容を考えるべきです。他の言語でごくごく当たり前に使っている配列変数なんてものでさえも，自分の実現したい内容を正確に把握できていなければ，それを実装することは困難を極めます。これは，プログラムのどの分野であっても同じ。プログラムを組むという仕事は，実際に命令を並べている時間よりも，仕様や実装方法を考えている時間の方が圧倒的に長くなるはずなのです。

さて，BMPでの配列変数ですが，以下のような書式にしようと思います。

```
$変数名[ 配列番号 ]
```

ちなみに配列変数というものですが，これは，「普通の変数の中に，多くのデータを入れられるようにしたもの」と考えてもらえればOKでしょう。例えば，

```
:$hensu = 1
```

というBMP命令では，$hensuという変数に，数値の1を入れるといった内容ですが，この$hensuには1種類のデータしか入れることができなく，

```
:$hensu = 3
```

とすると，元々入っていた1というデータは跡形もなく消えてしまい，残るのは後から入れた3だけです。配列変数の場合，一つの変数の中に複数のデータを入れることができるのですが，それを実現するために，変数の中をさらに細かくわけていきます。この，細かくわけた場所というのが，配列番号となるわけ。番号は通常，0番から始まる昇順の配列で表現されます。

さっきの例だと，同じ変数に1と3というデータを入れたい場合，

```
:$hensu[0] = 1
:$hensu[1] = 3
```

とすることで，$hensuの中に0番と1番という部屋を作り，その中に1と3というデータをそれぞれ入れることになります。配列番号がついていても，変数は変数なので，

```
$hensu[2] = $hensu[0]
```

のように，データを取り出すこともできます。

ですが，これだけでは，普通の変数を複数作るのと効果は何も変わりません。$hensu[0]や$hensu[1]の代わりに，$hensu0，$hensu1という変数を単独で用意することで同じことはできてしまいます。同じことができるのならば，わざわざ機能として用意する必要はありません。が，配列についてくる番号は，数値であるので変数で代用することも可能なのです。つまり，

```
:$a = 1
$hensu[$a] = 2
```

というように記述することができ，これは$hensu0のような別変数では実現できないことです。

とまぁ，配列変数というものを理解している人には，つまりBMPをここまで作ってきた人とっては，当たり前の話をしましたが，ここまで理解できているからこそ，自分で配列変数を作ることができるようになるのです。

配列変数を実装してみる

普通の変数と，配列変数の違いというのは，見た目には変数名の後ろに[]に囲われた配列番号がついてくるだけですが，内部構造はずいぶん変わってきます。データを入れる場所が複数あるということは，その分メモリを用意する必要がありますね。また，メモリ量を把握するためにも，その長さを管理しなくてはいけません。変数名は一緒ですので，一つの変数を「普通の変数か，それとも配列変数か」という形で拡張していくことにしましょう。となると，その変数タイプを管理する場所も必要になります。

前回までの変数の構造体は，以下のとおりでした。

≪リスト1≫ 今までの変数用構造体

```
typedef struct {
  char szName[32]; // 名前
  int  n;          // 値
} VALUESTRUCT;
```

この中には，変数名と，それに対応するデータという，最低限のものしかありません。配列変数をこれだけで表現するのは厳しすぎるので，この変数用の構造体を拡張することにします。

≪リスト2≫ 配列対応の変数用構造体

```
#define  VALTYPE_NUM      0  // 数値変数
#define  VALTYPE_EXTNUM   2  // 配列数値変数

typedef  struct {
  char  byType;          // 変数の種類(VALTYPE_XXXX)
  char  szName[32];      // 変数名
  int    n;              // 数値変数の場合，数値。配列変数の場合，
データ長
  void   *pExt;          // 配列変数時，データへのポインタ
} VALUESTRUCT;
```

増えたのは，変数の種類を示す値と，そのデータを格納する場所です。また，数値変数と配列変数の場合で，nの扱い方が変わってきます。数値変数の場合は，そのデータが直接入り，配列変数の場合は，現在の配列の大きさが入るのです。

そして，ついでに，変数の種類を定義しておきました。VALTYPE_xxxxというのが変数の種類です。数値に意味をもたせる場合，このように言葉で定義しておくことで，プログラム上の間違いが起きにくくなり，また修正がしやすくなりますね。

pExtには，配列用に確保したメモリへのポインタが入ります。例えば配列3番にデータを入れようとした場合，0,1,2,3のデータを入れる場所を確保する必要があります。この必要なメモリ容量は，sizeof(int) * (配列最大番号 + 1) で求めることができますので，メモリ確保命令malloc()でその分のメモリを確保し，その場所をpExtに保存しておくのです。これで，ひとつの変数にいくつものデータを格納するということが可能になります。

さて，実際のコードの修正は，変数処理系周り全体に及びます。前回までに用意した変数処理の関数（サブルーチン）と，その修正ポイントは以下のような感じなのです。

配列対応時に書き換えるもの

void ValueSysInit( );

・変数タイプを書き込む（VALTYPE_NUMで初期化）

・pExtに無効な値（NULL）を入れておく

void ValueSysRelease();

・配列用に確保したメモリも解放する

int ValueSysGetValueName(char *pStr, char *pBuf, int nBufSize);

・配列番号も含めた変数名を返すようにする

int ValueSysSearchValue (char *szName);

・変数が実在した場合，その配列番号も返すようにする

void ValueSysSetData (char *szName, int nData);

・配列番号が返ってきた場合，メモリを確保し配列にデータを入れる

int ValueSysGetData(char *szName);

・配列番号が指定されていた場合，配列側のメモリからデータを返す

記事上の都合で，実際のコードは，文字列変数周りも合わせた**リスト5**を確認していただくとして，こんな感じで変数システムの方だけ書き換えれば，実際のコマンド周りを書き換えなくても，あっさり配列変数対応になってしまうのです。

コード上のポイントは，配列が入っていようがなかろうが，使う側には意識させないで同じものとして処理しているところです。つまり，今までも「変数名」を得て，各関数に渡す必要がありましたが，この変数名の中に配列番号も含めて外に出しているのです。そして，変数システム内部では，渡された変数名に配列番号が含まれているかどうかを確認し，普通の変数の場合と，配列変数の場合の処理に分けています。変数システム内部はかなり変更がありますが，こうすることにより，そのシステムを使っている外側にはほとんど影響が出ないのです。

これは，配列変数であろうがなかろうが，出力すべきものは単一の数値データであるということが可能にさせています。もう一つの搭載機能である，文字列変数はこうは簡単にいきません。

文字列変数は外にも影響！

文字列変数も配列変数も，不定長のメモリを使うという点では同じようなデータの格納場所です。ということで，配列変数時に使ったpExtを，文字列変数でも用いていこうと思います。

配列変数の時とコード自体は変わりませんが，新しい変数用の構造体は次のようになります。

≪リスト3≫ 配列対応の変数用構造体

```
#define  VALTYPE_NUM      0  // 数値変数
#define  VALTYPE_STRING   1  // 文字列変数
#define  VALTYPE_EXTNUM   2  // 配列数値変数

typedef  struct {
  char  byType;          // 変数の種類(VALTYPE_XXXX)
  char  szName[32];      // 変数名
  int    n;              // 数値変数の場合，数値。文字列/配列変数
の場合，データ長
  void   *pExt;          // 文字列/配列変数時，データへのポインタ
} VALUESTRUCT;
```

配列変数と同じメモリを使うので，文字列変数を配列として用いることはできませんが，その代わりプログラムを非常に簡単にすることができます。さて，BMPで文字列変数はどのように使いましょうか。:PRINTという文字列出力命令で，""で囲うことで文字列指定をしていますので，この書式を踏襲し，

:$str = "文字列変数はこうするね"

このように，代入するものを""で囲うことで，文字列指定とします。

さてこうなると，困ってしまうのは，変数代入部分のプログラム。今まで「:$」と来た場合は，その後は変数名で，「=」に続くのは必ず数値，もしくは数値変数であるとしてきました。しかし，上の処理を行うとなると，「=」に続くデータが数値なのか文字列なのか判定しなくてはいけません。

これを行っているのが，**リスト5**のCheckDataType()です。やっていることは非常に簡単。0〜9の数値が続いていれば数値，「"」が来ていれば文字列，「$」があれば変数なので，その変数の種類を調べて返しています。変数の場合，配列変数も数値データであると返していますが，これは，配列変数から得られるデータもやはり個々の数値であるためです。

このような関数を作っておくことで，続くデータが数値の場合と，文字列の場合と，処理を変えることが可能です。実際，変数代入部分で，CheckDataType() を用いて，続くデータの種類によって処理を分岐させています。

数値データの場合は今までのものと同様ですが，問題は文字列データの場合です。変数システムも文字列変数に対応させなくてはいけません。というわけで，文字列変数対応用に，関数を以下の5つを追加しました。

文字列変数対応時に追加するもの

int CheckDataType (unsigned char *pSrc);

・続くデータが数値データか，文字データかを返します

void ValueSysSetString (char *szName, char *szData);

・変数に文字列を格納します

char *ValueSysGetString (char *szName);

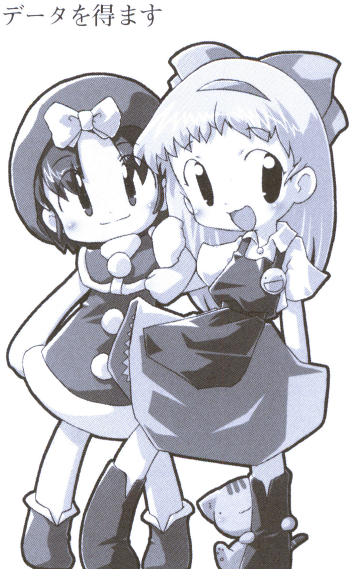
・文字列変数から文字列を得ます

int GetStringLenFromString (char *pSrc);

・続く文字列データがどれくらいの長さかを得ます

int GetStringFromString (char *pSrc, char *pDst);

・続く文字列データを得ます



今回のソースは頻繁にmalloc()やrealloc()を行う構成になっています。そのため，メモリの少ない環境では動作しない可能性もありますが，その場合ルールとして，文字列変数や配列で使えるメモリ量に制限をつけることで動くようになることもあります。

≪リスト4≫ macro.txt

```
/*
** BMPサンプルリスト
*/

:PRINT "ここから始まります"

:$data[0]=1
:$data[1]=3
:$data[2]=5
:$data[3]=6
:$data[4]=8
:$data[5]=9
:$data[6]=10
:$data[7]=12
:$data[8]=13
:$data[9]=14
:$str = "文字列も表示できるよ！"

// ループを組みます。
:$a = 0
#LOOP
:PRINT $data[$a]
:$a = $a + 1
:IF( $a < 10 ) { :GO 'LOOP' }
:PRINT $str
// ここがプログラム終端となります
#END
```

≪リスト5≫ script.c（差分）

```
/*
** Basicmagazine  Macro Processer
*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>


// 変数系処理 ////////////////////////////////////////////////////////////////////

#define  VALTYPE_NUM      0  // 数値変数
#define  VALTYPE_STRING   1  // 文字列変数
#define  VALTYPE_EXTNUM   2  // 配列数値変数

typedef  struct {
  char  byType;        // 変数の種類(VALTYPE_XXXX)
  char  szName[32];  // 変数名
  int    n;               // 数値変数の場合，数値。文字列/配列変数の場合，データ長
  void  *pExt;          // 文字列/配列変数時，データへのポインタ
} VALUESTRUCT;
VALUESTRUCT *pValue = NULL;
int nValueNum;
int nValueMaxNum;

// 変数システムの初期化
void ValueSysInit() {
  if ( !pValue ) {
    int i;
    nValueNum = 0;
    nValueMaxNum = 32;
    pValue = (VALUESTRUCT*)malloc( sizeof(VALUESTRUCT) * nValueMaxNum );
    for ( i = 0; i < nValueMaxNum; i++ ) {
      pValue[i].byType = VALTYPE_NUM;
```

つぎのページに続く

```
      pValue[i].pExt = NULL;
    }
  }
}


// 変数システムの解放
void ValueSysRelease() {
  if ( !pValue ) {
    int i;
    for ( i = 0; i < nValueMaxNum; i++ ) {
      if ( pValue[i].byType != VALTYPE_NUM ) {
        if ( pValue[i].pExt != NULL ) {
          free( pValue[i].pExt );
          pValue[i].pExt = NULL;
          pValue[i].n = 0;
        }
      }
    }
    free( pValue ); pValue = NULL;
  }
}

// 文字列から配列番号も含めた変数名を得る
int ValueSysGetValueName( char *pStr, char *pBuf, int nBufSize ) {
  int i = 0, j = 0;
  while( i < nBufSize ) {
    if ( ( 'A' <= pStr[i] && pStr[i] <= 'Z' )
     || ( 'a' <= pStr[i] && pStr[i] <= 'z' )
     || ( '0' <= pStr[i] && pStr[i] <= '9' )
     || ( pStr[i] == '_' ) ) {
      pBuf[j] = pStr[i];
      i++; j++;
    } else if ( pStr[i] == '[' ) {
      int nAns;
      i++; i += GetIntFromString( &pStr[i], &nAns );
      while( pStr[i] != ']' ) { i++; } i++;
      sprintf( &pBuf[j], "[%d]", nAns );
      j = strlen( pBuf );
    } else {
      break;
    }
  }
  pBuf[j] = '¥0';
  return i;
}

// その変数が実在するか確認，配列の場合，番号も格納する
int ValueSysSearchValue( char *szName, int *nExtNum ) {
  char szBuf[32];
  int i = 0;
  while( szName[i] != '¥0' && szName[i] != '[' ) {
    szBuf[i] = szName[i]; i++;
  }
  szBuf[i] = '¥0';
  *nExtNum = -1;
  for ( i = 0; i < nValueNum; i++ ) {
    if ( strcmp( szBuf, pValue[i].szName ) == 0 ) {
      if ( szName[strlen( szBuf )] == '[' ) {
        GetIntFromString( &szName[strlen( szBuf ) + 1], nExtNum );
      }
      return i;
    }
  }
  if ( szName[strlen( szBuf )] == '[' ) {
    GetIntFromString( &szName[strlen( szBuf ) + 1], nExtNum );
  }
  return -1;
}

// 変数に値を代入する
void ValueSysSetData( char *szName, int nData ) {
  int nExtNum;
  int n = ValueSysSearchValue( szName, &nExtNum );
  if ( n == -1 ) {
    char szBuf[32];
    int i = 0;
    while( szName[i] != '¥0' && szName[i] != '[' ) {
      szBuf[i] = szName[i]; i++;
    }
    szBuf[i] = '¥0';
    if ( nValueMaxNum == nValueNum ) {
      nValueMaxNum *= 2;
      pValue = (VALUESTRUCT*) realloc( pValue, sizeof(VALUESTRUCT) * nValueMaxNum );
    }
    strcpy( pValue[nValueNum].szName, szBuf );
    if ( nExtNum == -1 ) {
      pValue[nValueNum].n = nData;
      pValue[nValueNum].byType = VALTYPE_NUM;
    } else {
      pValue[nValueNum].n = nExtNum + 1;
      pValue[nValueNum].byType = VALTYPE_EXTNUM;
      pValue[nValueNum].pExt = malloc( sizeof( int ) * ( nExtNum + 1 ) );
      ((int*)pValue[nValueNum].pExt)[nExtNum] = nData;
    }
    nValueNum++;
  } else {
    if ( nExtNum == -1 ) {
      pValue[n].n = nData;
    } else {
      if ( pValue[n].n <= nExtNum ) {
        pValue[n].pExt = realloc( pValue[n].pExt, sizeof( int ) * ( nExtNum + 1 ) );
        pValue[n].n = nExtNum + 1;
      }
      ((int*)pValue[n].pExt)[nExtNum] = nData;
    }
  }
}

// 変数の値を得る
int ValueSysGetData( char *szName ) {
  int nExtNum;
  int n = ValueSysSearchValue( szName, &nExtNum );
  if ( n == -1 ) {
    return 0;
  }
  if ( nExtNum == -1 ) {
    return pValue[n].n;
  } else {
    return ((int*)pValue[n].pExt)[nExtNum];
  }
}

// データの種類を調べる
int CheckDataType( unsigned char *pSrc ) {
  unsigned long dwCnt = 0;
  dwCnt += SkipSpaceFromMemory( &pSrc[dwCnt] );
  if ( pSrc[dwCnt] == '"' ) { return VALTYPE_STRING; }
  if ( pSrc[dwCnt] == '$' ) {
    char szValueName[32];
    int n, nExtNum;
    ValueSysGetValueName( (char*)&pSrc[dwCnt+1], szValueName, sizeof(szValueName) );
    n = ValueSysSearchValue( szValueName, &nExtNum );
    if ( n != -1 ) {
      switch( pValue[n].byType ) {
        case VALTYPE_NUM:
        case VALTYPE_EXTNUM:
          return VALTYPE_NUM;
        case VALTYPE_STRING:
          return VALTYPE_STRING;
      }
    }
```

→ つぎのページに続く

```
  }
  return VALTYPE_NUM;
}

// 文字列変数に文字列を格納（文字列変数は配列に対応していません）
void ValueSysSetString( char *szName, char *szData ) {
  int nExtNum;
  int n = ValueSysSearchValue( szName, &nExtNum );
  if ( n == -1 ) {
    char szBuf[32];
    int i = 0;
    while( szName[i] != '¥0' && szName[i] != '[' ) {
      szBuf[i] = szName[i]; i++;
    }
    szBuf[i] = '¥0';
    if ( nValueMaxNum == nValueNum ) {
      nValueMaxNum *= 2;
      pValue = (VALUESTRUCT*) realloc( pValue, sizeof(VALUESTRUCT) * nValueMaxNum );
    }
    strcpy( pValue[nValueNum].szName, szBuf );
    pValue[nValueNum].n = strlen( szData );
    pValue[nValueNum].byType = VALTYPE_STRING;
    pValue[nValueNum].pExt = malloc( pValue[nValueNum].n + 1 );
    strcpy( pValue[nValueNum].pExt, szData );
    nValueNum++;
  } else {
    pValue[n].n = strlen( szData );
    if ( pValue[n].pExt != NULL ) {
      free( pValue[n].pExt );
    }
    pValue[n].pExt = malloc( pValue[n].n + 1 );
    strcpy( pValue[n].pExt, szData );
  }
}

// 文字列変数のデータを得る
char *ValueSysGetString( char *szName ) {
  int nExtNum;
  int n = ValueSysSearchValue( szName, &nExtNum );
  if ( n == -1 ) {
    return 0;
  }
  return pValue[n].pExt;
}



//////////////////////////////////////////////////////////////////////////////////////////////////
////////////
// メモリから，その文字列がどれくらいのメモリを必要とするかを得る
int GetStringLenFromString( char *pSrc ) {
  unsigned long dwCnt = 0;
  int nLen = 0;
  dwCnt += SkipSpaceFromMemory( &pSrc[dwCnt] );
  if ( pSrc[dwCnt] == '"' ) {
    dwCnt++;
    while( pSrc[dwCnt] != '"' ) {
      if ( IsKanji( pSrc[dwCnt] ) ) { dwCnt += 2; nLen += 2; }
      else { dwCnt++; nLen++; }
    }
    return nLen;
  }
  if ( pSrc[dwCnt] == '$' ) {
    char szValueName[32];
    int n, nExtNum;
    ValueSysGetValueName( (char*)&pSrc[dwCnt+1], szValueName, sizeof(szValueName) );
    n = ValueSysSearchValue( szValueName, &nExtNum );
    return pValue[n].n;
  }
  return 0;
}
```

```
// 文字列から文字列を得る
int GetStringFromString( char *pSrc, char *pDst ) {
  unsigned long dwCnt = 0;
  int nLen = 0;
  dwCnt += SkipSpaceFromMemory( &pSrc[dwCnt] );
  if ( pSrc[dwCnt] == '"' ) {
    dwCnt++;
    while( pSrc[dwCnt] != '"' ) {
      if ( IsKanji( pSrc[dwCnt] ) ) {
        pDst[nLen] = pSrc[dwCnt];
        pDst[nLen+1] = pSrc[dwCnt+1];
        dwCnt += 2; nLen += 2;
      } else {
        pDst[nLen] = pSrc[dwCnt];
        dwCnt++; nLen++;
      }
    }
    pDst[nLen] = '¥0';
  }
  if ( pSrc[dwCnt] == '$' ) {
    char szValueName[32];
    int n, nExtNum;
    dwCnt += ValueSysGetValueName( (char*)&pSrc[dwCnt+1], szValueName,
sizeof(szValueName) );
    n = ValueSysSearchValue( szValueName, &nExtNum );
    strcpy( pDst, pValue[n].pExt );
  }
  return dwCnt;
}

//////////////////////////////////////////////////////////////////////////////////////////////////
//////////

// プログラムはここから始まる
void main() {
  FILE *fp; int nSize;
  unsigned char *p, *pWork;
  int nPnt = 0;
  // :GSサブルーチンジャンプは64階層まで可能
  int nGosubStackCnt = 0, nGosubStack[ 64 ];

  // 初期化
  ValueSysInit();

  // マクロファイルをメモリに読み込む
  fp = fopen( "macro.txt", "rb" );
  fseek( fp, 0, SEEK_END ); nSize = ftell( fp ); fseek( fp, 0, SEEK_SET );
  p = ( unsigned char* )malloc( nSize + 1);
  fread( p, 1, nSize, fp );
  fclose( fp );

  // 読んだデータの終端に¥0をつける
  p[ nSize ] = '¥0';

  // compiler //////////////////////////////////////////////////////////////////
  pWork = Compiler( p, &nSize );
  free( p ); p = pWork;

  // interpriter //////////////////////////////////////////////////////////////////
  while( nPnt < nSize ) {
    if ( IsKanji( p[nPnt] ) ) {
      // SHIFT-JIS漢字コードなので2byte進める
      nPnt += 2;
    } else {
      if ( p[nPnt] == ':' ) {
        // コマンドマーク有
        nPnt++;
        if ( p[nPnt] == '$' ) {
          //MKADD
          // 変数代入
```

→ つぎのページに続く

```
        char szValueName[32];
        int nValue, nType;
        nPnt++;
        nPnt += ValueSysGetValueName( (char*)&p[nPnt], szValueName,
sizeof(szValueName) );
        while( p[nPnt] != '=' ) { nPnt++; } nPnt++;
        nType = CheckDataType( &p[nPnt] );
        if ( nType == VALTYPE_NUM ) {
          nPnt += GetIntFromString( (char*)&p[nPnt], &nValue );
          ValueSysSetData( szValueName, nValue );
        } else {
          char *pBuf;
          int nLength;
          nLength = GetStringLenFromString( (char*)&p[nPnt] );
          pBuf = malloc( nLength + 1 );
          nPnt += GetStringFromString( (char*)&p[nPnt], pBuf );
          ValueSysSetString( szValueName, pBuf );
          free( pBuf );
        }
      } else {
        // コマンド処理
        char szCmd[32]; int i = 0;
        while( ( 'A' <= p[nPnt] && p[nPnt] <= 'Z' )
          || ( 'a' <= p[nPnt] && p[nPnt] <= 'z' ) ) {
          szCmd[i] = p[nPnt]; nPnt++; i++;
        }
        szCmd[i] = '¥0';
        if ( strcmpi( szCmd, "PRINT" ) == 0 ) {
          // MKADD
          int nAns, nType;
          nPnt += SkipSpaceFromMemory( &p[nPnt] );
          nType = CheckDataType( &p[nPnt] );
          if ( nType == VALTYPE_NUM ) {
            // 数値表示
            nPnt += GetIntFromString( (char*)&p[nPnt], &nAns );
            printf( "%d¥n", nAns );
          } else {
            // 文字列表示
            char *pBuf;
            int nLength;
            nLength = GetStringLenFromString( (char*)&p[nPnt] );
            pBuf = malloc( nLength + 1 );
            nPnt += GetStringFromString( (char*)&p[nPnt], pBuf );
            printf( "%s¥n", pBuf );
            free( pBuf );
          }
        } else if ( strcmpi( szCmd, "IF" ) == 0 ) {
          // 前回と変わらないので省略
        } else if ( strcmpi( szCmd, "GO" ) == 0 ) {
          // 前回と変わらないので省略
        } else if ( strcmpi( szCmd, "GS" ) == 0 ) {
          // 前回と変わらないので省略
        } else if ( strcmpi( szCmd, "RT" ) == 0 ) {
          // 前回と変わらないので省略
        }
      }
    } else if ( p[nPnt] == '}' ) {
      // 前回と変わらないので省略
    } else {
      // コマンドでないので1byte進める
      nPnt++;
    }
  }
}

// 解放
ValueSysRelease();
free( p );
}
```

▲長いので，前回と変わらない処理は省略しました

ついに完成！　そして，最終回

5回に渡って作ってきました「オリジナル・プログラム言語の制作！」ベーマガ・マクロプロセッサも，今回で所期の目標は達成しました。

今の仕様でも，十分，プログラムの「考え方」を表現するだけの能力はあると思います。BMPをベースに，ここから，絵や音周りを整備して，ゲームに特化した言語に仕立て上げても良いですし，ネットワークに対応させてBMPでCGI，なんていうのも夢のある話で面白いかもしれません。本誌掲載のソースはいくらでも使ってもらって構いませんので，あなたの目的に合わせて，さまざまな言語を作って，楽しんでみてください。

また，ボクの方でもWindows用でBMPをベースにした言語を作っています。Windows用としては，まだまだ機能は少ないけれど，この号が出る頃には本誌のホームページで紹介できるように考えていますので，良かったら見てみてくださいね。あ，そうそう，今回のBMPソースもホームページからダウンロードできますので，今までの記事をうまく合わせられない人は，そちらをご利用くださいませ。

さて。2年半という長い間，連載をさせて頂きました「もっと身近に」ですが，今回でフィナーレとなります。ゲームプログラミングといえど，機種にとらわれず，重要なのは「考え方」なんだよ，というのをメインに書いてきました。実際にゲームを作る際に，「そういえば，『もっと身近に』でやり方書いてあったなぁ」なんて思い返して見てもらえれば嬉しいですね。

それでは，また近いうちに，別の記事で会いましょう。ご愛読ありがとうございました。



（無　料）
フリーソフト
HSP 完全マスター
Hot Soup Processor

楽々3Dプログラミング！

第5回

監修：おにたま

講師：悠黒喧史

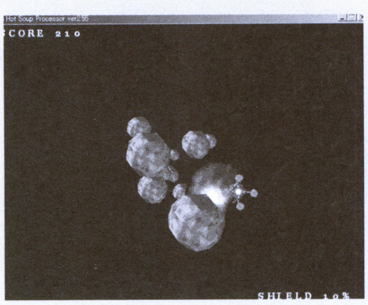


いよいよ実践篇である。甘えは許されない。失敗は即，死につながると知れ。NINJAの掟は無情なのだ。3Dゲイム如きが何ほどのものか。ふ，たわいない。

「MXモデルでゲーム制作(2)」

今回のポイント

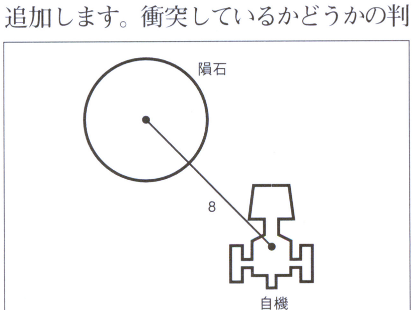
コリジョンの設定とUVアニメーション



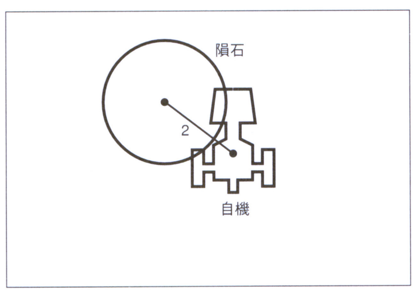
コリジョン(当たり判定)の設定

オブジェクト同士の衝突判定

自機と隕石の衝突判定をプログラムに追加します。衝突しているかどうかの判定は，オブジェクト同士の距離によって判断します。例えば，判定距離を3と指定した場合は，実際のオブジェクト同士の距離が3以下になったとき，衝突していると判定されます。



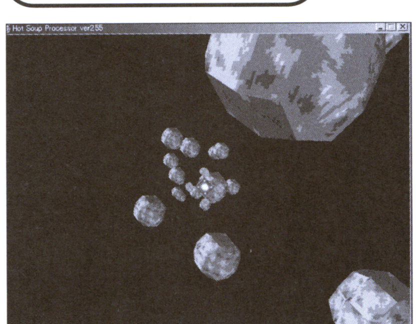
《第1図》衝突してない(距離=8)



《第2図》衝突している(距離=2)

自機と隕石の衝突判定（リスト3）

リスト3の実行画面

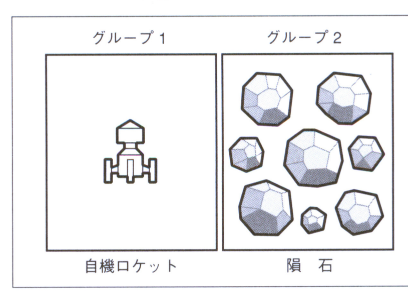


▲カーソルキーで上下左右に移動します。Escでプログラムを終了します。

コリジョン処理の対象を設定

衝突判定を行なうには，まず，setcoli命令でコリジョン処理の対象となるオブジェクトにグループ番号を設定します。ここでは，自機オブジェクトをグループ番号1に設定し，そのグループ番号1に衝突するグループの番号を2としています。そして，隕石の発生を行なう処理部分で，作成される隕石オブジェクトのグループ番号をすべて2に設定しています。

グループの設定



《第3図》グループの設定

実際の衝突判定

設定されたグループ番号のコリジョン情報をもとに，getcoli命令で，実際の衝突判定を行ないます。判定したいオブジェクトのID番号と，衝突する距離を指定すると，現在，衝突しているオブジェクトのID番号を返してくれます。衝突していない場合は，－1が返ります。

《リスト3》自機と隕石の衝突判定のプログラム・リスト

```
#include "hgimg.as"

        hgini                                           ; HGIMGの初期化
; ------------------------------------------------------------------
        mxload "myship" :mdmy=stat                      ; 自機ﾓﾃﾞﾙﾌｧｲﾙ読込み
        mxload "asteroid" :mdas=stat                    ; 隕石ﾓﾃﾞﾙﾌｧｲﾙ読込み
; ------------------------------------------------------------------
        regobj obmy,mdmy,OBJ_FLIP|OBJ_BORDER            ; 自機ｵﾌﾞｼﾞｪｸﾄ作成
        selang obmy :objsetf3 -3.1416f,0.0f,0.0f    ; 角度ｾｯﾄ
        selpos obmy :objset3 0,0,50                     ; 座標ｾｯﾄ
        setcoli obmy,1,2                                ; ｺﾘｼﾞｮﾝのｸﾞﾙｰﾌﾟNo.設定
; ------------------------------------------------------------------
        selcpos :objset3 0,0,80                         ; ｶﾒﾗ座標
; ------------------------------------------------------------------
        setborder 20.0f,20.0f,160.0f                    ; ｵﾌﾞｼﾞｪｸﾄの有効ｴﾘｱを設定
; ------------------------------------------------------------------
        gsel 0
*main
        getcoli ob,obmy,2.5f                            ; ｺﾘｼﾞｮﾝ判定
        if ob!-1 :delobj ob                             ; 衝突してる隕石ｵﾌﾞｼﾞｪｸﾄを消去

        stick ky,255                                    ; ｷｰｽｷｬﾝ

        selpos obmy :objgetfv po                        ; po ← 自機MOC座標

        if ky&1 :fsub po.0,0.2f                         ; X軸移動
        if ky&4 :fadd po.0,0.2f                         ;
        if ky&2 :fsub po.1,0.2f                         ; Y軸移動
        if ky&8 :fadd po.1,0.2f                         ;

        objsetfv po                                     ; 自機MOC座標 ← po

        rnd r,9
        if r=0 { regobj obas,mdas,OBJ_MOVE|OBJ_BORDER   ; 隕石ｵﾌﾞｼﾞｪｸﾄ作成
                rnd x,20 :rnd y,20                      ;
                selpos obas :objset3 x-10,y-10,-80      ; 座標ｾｯﾄ
                seldir obas :objset3 0,0,1              ; 移動量ｾｯﾄ
                setcoli obas,2                          ; ｺﾘｼﾞｮﾝのｸﾞﾙｰﾌﾟNo.設定
        }

        hgdraw :hgsync 15                               ; 描画 / ｳｪｲﾄ

        if ky!128 :goto *main                           ; [Esc]を押すまでﾙｰﾌﾟ

        hgbye                                           ; HGIMGを解放
```

衝突した隕石を消去

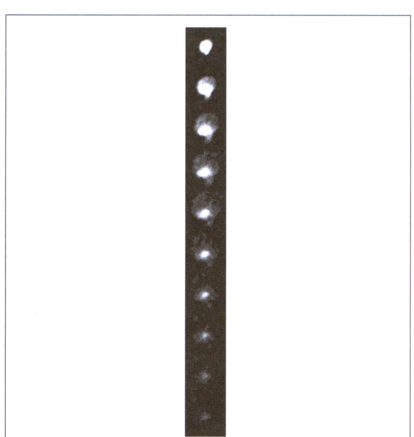
ここでは，自機オブジェクト(グループ1)に衝突している隕石オブジェクト(グループ2)のID番号が返りますので，返り値が−1でなかった場合には，**delobj**命令でオブジェクト消去しています。

使用した命令

- setcoli　コリジョンの対象オブジェクトをグループ設定する
- getcoli　オブジェクトの衝突判定を行なう
- delobj　指定されたオブジェクトを消去する

爆発アニメーション

隕石が自機にぶつかったとき，ただ消



▲隕石の爆発テクスチャ

えるだけでは，あまりに演出が寂しいので，爆発アニメーションをプログラムに追加しましょう。爆発アニメーションは，プレート(平面板)モデルを使った**UVアニメーション**という方法で表現します。

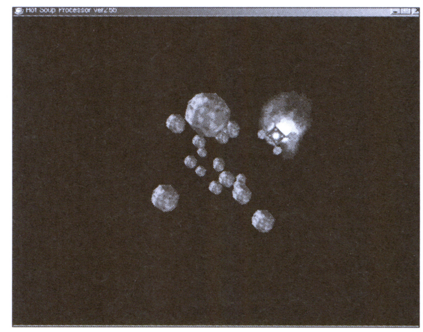
隕石の爆発テクスチャ

bom_as.bmp

64×640ドット

隕石の爆発アニメ(リスト4)

リスト4の実行画面



▲カーソルキーで上下左右に移動します。Escでプログラムを終了します。

まず，爆発アニメーションのテクスチャを貼り付けるプレートモデルを登録しておきます。プレートモデルの作り方は，これまでと同じですが，最後に**uvanim**命令を付け加えて**UVアニメーション**の設定を行ないます。

UVアニメーションは，指定したテクスチャのY座標を一定時間ごとに縦にずらしていくことで実現しますので，元のテクスチャには縦方向にアニメーションパターンが連続して描かれている必要があります。

爆発オブジェクトの作成

衝突している隕石オブジェクトのID番号から，消去前にMOC座標を得ておきます。**delobj**命令で隕石オブジェクトを消去し，そのMOC座標に爆発オブジェクトを作成します。

爆発オブジェクトはプレートモデルから作成しているので，常にカメラの正面を向くように**OBJ_XFRONT**属性を指定しています。また，爆発アニメーションが終了したら自動的に消去されるように**OBJ_UVANIM_1SHOT**属性を指定しています。

《リスト4》隕石の爆発アニメのプログラム・リスト

```
#include "hgimg.as"

	hgini				; HGIMGの初期化
; ---------------------------------------------------------------------------
	setsizef 8.0f,8.0f		; ﾌﾟﾚｰﾄXYｻｲｽﾞ
	setuv 0,0,63,63			; ﾃｸｽﾁｬUV座標
	addplate mdex,1			; ﾌﾟﾚｰﾄﾓﾃﾞﾙ作成
	texload "bom_as.bmp"		; 爆発ﾃｸｽﾁｬ読込み
	uvanim mdex,5,10,1		; ｱﾆﾒｰｼｮﾝ設定
; ---------------------------------------------------------------------------
	mxload "myship" :mdmy=stat	; 自機ﾓﾃﾞﾙﾌｧｲﾙ読込み
	mxload "asteroid" :mdas=stat	; 隕石ﾓﾃﾞﾙﾌｧｲﾙ読込み
; ---------------------------------------------------------------------------
	regobj obmy,mdmy,OBJ_FLIP|OBJ_BORDER	; 自機ｵﾌﾞｼﾞｪｸﾄ作成
	selang obmy :objsetf3 -3.1416f,0.0f,0.0f	; 角度ｾｯﾄ
	selpos obmy :objset3 0,0,50	; 座標ｾｯﾄ
	setcoli obmy,1,2		; ｺﾘｼﾞｮﾝのｸﾞﾙｰﾌﾟNo.設定
; ---------------------------------------------------------------------------
	selcpos :objset3 0,0,80		; ｶﾒﾗ座標
; ---------------------------------------------------------------------------
	setborder 20.0f,20.0f,160.0f	; ｵﾌﾞｼﾞｪｸﾄの有効ｴﾘｱを設定
; ---------------------------------------------------------------------------
	gsel 0
*main
	getcoli ob,obmy,2.5f		; ｺﾘｼﾞｮﾝ判定
	if ob!-1 { selpos ob :objgetfv po
		delobj ob			; 衝突隕石を消去
		regobj obex,mdex,OBJ_XFRONT|OBJ_UVANIM_1SHOT	; 爆発ｵﾌﾞｼﾞｪｸﾄの作成
		selpos obex :objsetfv po
	}

	stick ky,255			; ｷｰｽｷｬﾝ

	selpos obmy :objgetfv po	; po ← 自機MOC座標

	if ky&1 :fsub po.0,0.2f		; X軸移動
	if ky&4 :fadd po.0,0.2f		;
	if ky&2 :fsub po.1,0.2f		; Y軸移動
	if ky&8 :fadd po.1,0.2f		;

	objsetfv po			; 自機MOC座標 ← po

	rnd r,9
	if r=0 { regobj obas,mdas,OBJ_MOVE|OBJ_BORDER; 隕石ｵﾌﾞｼﾞｪｸﾄ作成
		rnd x,20 :rnd y,20	;
		selpos obas :objset3 x-10,y-10,-80	; 座標ｾｯﾄ
		seldir obas :objset3 0,0,1	; 移動量ｾｯﾄ
		setcoli obas,2		; ｺﾘｼﾞｮﾝのｸﾞﾙｰﾌﾟNo.設定
	}

	hgdraw :hgsync 15		; 描画 / ｳｪｲﾄ

	if ky!128 :goto *main		; [Esc]を押すまでﾙｰﾌﾟ

	hgbye				; HGIMGを解放
```

使用した命令

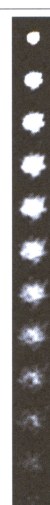
uvanim　　モデルにUVアニメーション設定をする

ゲームオーバーのルール

隕石が自機に衝突するごとにシールドエネルギーが減り，0のときに衝突するとゲームオーバーになるというゲームのルールと，自機の爆発アニメーションをプログラムに追加します。

自機の爆発テクスチャ

bom_my.bmp

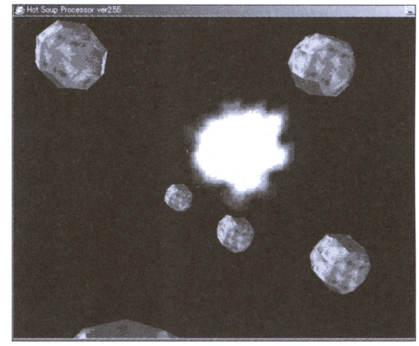


▲自機の爆発テクスチャ

80×960ドット

ゲームルールの追加（リスト5）

リスト5の実行画面



▲カーソルキーで上下左右に移動します。Escでプログラムを終了します。

隕石の爆発処理と同様，自機の爆発アニメーション用のプレートモデルを登録しておきます。

変数enを自機のシールドエネルギーとし，隕石が衝突するごとに減らしています。衝突時に変数enが0であれば，自機オブジェクトを消去し，自機の爆発オブジェクトを同じ場所に作成します。このとき，変数ovを自機の爆発アニメーションが終了するまでのカウンタとしてセットしています。変数ovの値を減算しながら0になるまで待ち，0になったらプログラムを停止しています。

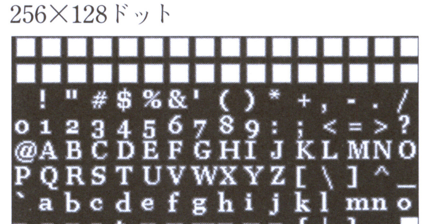
文字表示と効果音を追加しよう

スコア処理や，シールドエネルギーの表示，爆発の効果音などをプログラムに追加します。

文字のフォント画像

font.bmp

256×128ドット



▲文字のフォント画像

隕石の爆発音

dmg.wav

《リスト5》ゲームルールの追加プログラム・リスト

```
#include "hgimg.as"

        hgini                                                   ; HGIMGの初期化
; ------------------------------------------------------------------------------
        setsizef 15.0f,15.0f                                    ; ﾌﾟﾚｰﾄXYｻｲｽﾞ
        setuv 0,0,79,79                                         ; ﾃｸｽﾁｬUV座標
        addplate mdez,1                                         ; ﾌﾟﾚｰﾄﾓﾃﾞﾙ作成
        texload "bom_my.bmp"                                    ; 自機爆発ﾃｸｽﾁｬ読込み
        uvanim mdez,5,12,1                                      ; ｱﾆﾒｰｼｮﾝ設定
; ------------------------------------------------------------------------------
        setsizef 8.0f,8.0f                                      ; ﾌﾟﾚｰﾄXYｻｲｽﾞ
        setuv 0,0,63,63                                         ; ﾃｸｽﾁｬUV座標
        addplate mdex,1                                         ; ﾌﾟﾚｰﾄﾓﾃﾞﾙ作成
        texload "bom_as.bmp"                                    ; 隕石爆発ﾃｸｽﾁｬ読込み
        uvanim mdex,5,10,1                                      ; ｱﾆﾒｰｼｮﾝ設定
; ------------------------------------------------------------------------------
        mxload "myship" :mdmy=stat                              ; 自機ﾓﾃﾞﾙﾌｧｲﾙ読込み
        mxload "asteroid" :mdas=stat                            ; 隕石ﾓﾃﾞﾙﾌｧｲﾙ読込み
; ------------------------------------------------------------------------------
        regobj obmy,mdmy,OBJ_FLIP|OBJ_BORDER                    ; 自機ｵﾌﾞｼﾞｪｸﾄ作成
        selang obmy :objsetf3 -3.1416f,0.0f,0.0f        ; 角度ｾｯﾄ
        selpos obmy :objset3 0,0,50                             ; 座標ｾｯﾄ
        setcoli obmy,1,2                                        ; ｺﾘｼﾞｮﾝのｸﾞﾙｰﾌﾟNo.設定
; ------------------------------------------------------------------------------
        selcpos :objset3 0,0,80                                 ; ｶﾒﾗ座標
; ------------------------------------------------------------------------------
        setborder 20.0f,20.0f,160.0f                            ; ｵﾌﾞｼﾞｪｸﾄの有効ｴﾘｱを設定
; ------------------------------------------------------------------------------
        en=100                                                  ; ｼｰﾙﾄﾞ値

        gsel 0
*main
        if ov>0 :ov- :if ov=0 :stop                             ; 自機爆発ｶｳﾝﾄ & 爆発ｱﾆﾒｰｼｮﾝ終了待ち

        getcoli ob,obmy,2.5f                                    ; ｺﾘｼﾞｮﾝ判定
        if ob!-1 { selpos ob :objgetfv po
                delobj ob                                       ; 衝突隕石を消去
                regobj obex,mdex,OBJ_XFRONT|OBJ_UVANIM_1SHOT    ; 隕石爆発ｵﾌﾞｼﾞｪｸﾄ作成
                selpos obex :objsetfv po
                if en=0 { selpos obmy :objgetfv po
                        regobj obez,mdez,OBJ_XFRONT|OBJ_UVANIM_1SHOT  ; 自機爆発ｵﾌﾞｼﾞｪｸﾄ作成
                        selpos obez :objsetfv po
                        delobj obmy :ov=200                     ; 自機消去 / 爆発ｱﾆﾒｰｼｮﾝのｶｳﾝﾀ
                } else :en-=10                                  ; ｼｰﾙﾄﾞｴﾈﾙｷﾞｰ減
        }

        stick ky,255                                            ; ｷｰｽｷｬﾝ

        selpos obmy :objgetfv po                                ; MOC座標を変数po.0～2に得る

        if ky&1 :fsub po.0,0.2f                                 ; X軸移動
        if ky&4 :fadd po.0,0.2f                                 ;
        if ky&2 :fsub po.1,0.2f                                 ; Y軸移動
        if ky&8 :fadd po.1,0.2f                                 ;

        objsetfv po                                             ; 変数po.0～2の内容をMOC座標に戻す

        rnd r,9
        if r=0 { regobj obas,mdas,OBJ_MOVE|OBJ_BORDER  ; 隕石ｵﾌﾞｼﾞｪｸﾄ作成
                rnd x,20 :rnd y,20                              ;
                selpos obas :objset3 x-10,y-10,-80              ; 座標ｾｯﾄ
                seldir obas :objset3 0,0,1                      ; 移動量ｾｯﾄ
                setcoli obas,2                                  ; ｺﾘｼﾞｮﾝのｸﾞﾙｰﾌﾟNo.設定
        }

        hgdraw :hgsync 15                                       ; 描画 / ｳｪｲﾄ

        if ky!128 :goto *main                                   ; [Esc]を押すまでﾙｰﾌﾟ

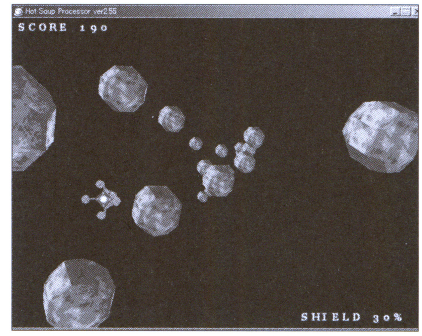
        hgbye                                                   ; HGIMGを解放
```

自機の爆発音

explos.wav

文字と音の追加(リスト6)

リスト6の実行画面



▲カーソルキーで上下左右に移動します。Escでプログラムを終了します。

スコア処理

変数scをスコアの値として用意し，変数tでメインループ100回ごとにスコア加算しています。

効果音

爆発音は，sndload命令でWAVデータをあらかじめ読み込んでおき，衝突時にsnd命令で鳴らしています。

※効果音に使うWAVファイル

効果音データは，インターネット上でフリーで公開されているものを利用するか，自分で録音して作成しましょう。また，著作権フリーのサウンド素材集なども販売されています。本誌のホームページにも置いてありますので，ぜひ活用してください。

スコア，シールドエネルギーの表示

setfont命令は，fprt命令で表示するためのフォントを設定します。直後にtexload命令で読み込まれた画像がフォントになります。この文字表示用のフォント画像は，横16文字分で改行するアスキーコード順に並べて作成します。

fprt命令で，スコアとシールドエネルギーの値を画面に表示しています。HGIMGでは，フォント表示は必ずhgdraw命令と，hgsync命令の間に使用する決まりになっています。

《リスト6》文字と音の追加プログラム・リスト

```
#include "hgimg.as"

        hgini                                                   ; HGIMGの初期化
; ---------------------------------------------------------------------------------------
        setfont 16,16,16,1 :texload "font.bmp"                  ; ﾌｫﾝﾄﾃｸｽﾁｬ登録
; ---------------------------------------------------------------------------------------
        setsizef 15.0f,15.0f                                    ; ﾌﾟﾚｰﾄXYｻｲｽﾞ
        setuv 0,0,79,79                                         ; ﾃｸｽﾁｬUV座標
        addplate mdez,1                                         ; ﾌﾟﾚｰﾄﾓﾃﾞﾙ作成
        texload "bom_my.bmp"                                    ; 自機爆発ﾃｸｽﾁｬ読込み
        uvanim mdez,5,12,1                                      ; ｱﾆﾒｰｼｮﾝ設定
; ---------------------------------------------------------------------------------------
        setsizef 8.0f,8.0f                                      ; ﾌﾟﾚｰﾄXYｻｲｽﾞ
        setuv 0,0,63,63                                         ; ﾃｸｽﾁｬUV座標
        addplate mdex,1                                         ; ﾌﾟﾚｰﾄﾓﾃﾞﾙ作成
        texload "bom_as.bmp"                                    ; 隕石爆発ﾃｸｽﾁｬ読込み
        uvanim mdex,5,10,1                                      ; ｱﾆﾒｰｼｮﾝ設定
; ---------------------------------------------------------------------------------------
        mxload "myship" :mdmy=stat                              ; 自機ﾓﾃﾞﾙﾌｧｲﾙ読込み
        mxload "asteroid" :mdas=stat                            ; 隕石ﾓﾃﾞﾙﾌｧｲﾙ読込み
; ---------------------------------------------------------------------------------------
        regobj obmy,mdmy,OBJ_FLIP|OBJ_BORDER                    ; 自機ｵﾌﾞｼﾞｪｸﾄ作成
        selang obmy :objsetf3 -3.1416f,0.0f,0.0f        ; 角度ｾｯﾄ
        selpos obmy :objset3 0,0,50                             ; 座標ｾｯﾄ
        setcoli obmy,1,2                                        ; ｺﾘｼﾞｮﾝのｸﾞﾙｰﾌﾟNo.設定
; ---------------------------------------------------------------------------------------
        selcpos :objset3 0,0,80                                 ; ｶﾒﾗ座標
; ---------------------------------------------------------------------------------------
        setborder 20.0f,20.0f,160.0f                            ; ｵﾌﾞｼﾞｪｸﾄの有効ｴﾘｱを設定
; ---------------------------------------------------------------------------------------
        sndload "dmg.wav",0                                     ; 隕石衝突音
        sndload "explos.wav",1                                  ; 自機爆発音

        en=100 :sc=0 :t=0                                       ; ｼｰﾙﾄﾞ値 / ｽｺｱ

        gsel 0
*main
        if ov>0 :ov- :if ov=0 :stop                             ; 自機の爆発ｱﾆﾒｰｼｮﾝ終了待ち

        getcoli ob,obmy,2.5f                                    ; ｺﾘｼﾞｮﾝ判定
        if ob!-1 { snd 0 :selpos ob :objgetfv po
                delobj ob                                       ; 衝突隕石を消去
                regobj obex,mdex,OBJ_XFRONT|OBJ_UVANIM_1SHOT ; 隕石爆発ｵﾌﾞｼﾞｪｸﾄ作成
                selpos obex :objsetfv po
                if en=0 { snd 1 :selpos obmy :objgetfv po
                        delobj obmy                             ; 自機を消去
                        regobj obez,mdez,OBJ_XFRONT|OBJ_UVANIM_1SHOT ; 自機爆発ｵﾌﾞｼﾞｪｸﾄ作成
                        selpos obez :objsetfv po
                        ov=200                                  ; 爆発ｱﾆﾒｰｼｮﾝ終了待ちのｶｳﾝﾀ
                } else :en-=10                                  ; ｼｰﾙﾄﾞｴﾈﾙｷﾞｰ減
        }

        stick ky,255                                            ; ｷｰｽｷｬﾝ

        selpos obmy :objgetfv po                                ; MOC座標を変数po.0～2に得る

        if ky&1 :fsub po.0,0.2f                                 ; X軸移動
        if ky&4 :fadd po.0,0.2f                                 ;
        if ky&2 :fsub po.1,0.2f                                 ; Y軸移動
        if ky&8 :fadd po.1,0.2f                                 ;

        objsetfv po                                             ; 変数po.0～2の内容をMOC座標に戻す

        rnd r,9
        if r=0 { regobj obas,mdas,OBJ_MOVE|OBJ_BORDER ; 隕石ｵﾌﾞｼﾞｪｸﾄ作成
                rnd x,20 :rnd y,20                              ;
                selpos obas :objset3 x-10,y-10,-80              ; 座標ｾｯﾄ
                seldir obas :objset3 0,0,1                      ; 移動量ｾｯﾄ
                setcoli obas,2                                  ; ｺﾘｼﾞｮﾝのｸﾞﾙｰﾌﾟNo.設定
        }

        t+ :if t¥100=0 :sc+=10                                  ; ｽｺｱ加算

        hgdraw                                                  ; 描画

        fprt "SCORE "+sc,8,8                                    ; ｽｺｱ表示
        fprt "SHIELD "+en+"%",448,456                           ; ｼｰﾙﾄﾞ表示

        hgsync 15                                               ; ｳｪｲﾄ

        if ky!128 :goto *main                                   ; [Esc]を押すまでﾙｰﾌﾟ

        hgbye                                                   ; HGIMGを解放
```

使用した命令

setfont　フォントの登録
fprt　指定文字列を登録フォントで表示

さて，これで一応，ゲームとして最小限必要な部分は，ほぼ完成しました。しかし，ゲームは，遊べる部分さえ出来れば完璧，というものではありません。より面白くなるよう，楽しんでもらえる工夫を加えてゆくことが，とても重要なポイントなのです。

最終回となる次の講座では，ゲームとしてのクオリティをさらにアップするための改良を加え，アドバイスしてゆきます。

HGIMG-FAQ

質問 

HGIMGには，DirectXを使うバージョンがあるようですが，どうやって使うのですか？

答え 

DirectXを使うHGIMGは"hgimgx.dll"で，スクリプトでは，最初に次の1行を入れます。

#include "hgimgx.as"

"hgimgx.dll"はDirectXを使用しますが，DirectX自身がビデオボードとの相性に問題を起こす場合もあります。

不特定多数のパソコンでの安定動作を重視する場合は，Windowsの標準グラフィックスシステムであるGDI(Graphical Device Interface)を利用する"hgimg.dll"を使うと良いでしょう。

爆発アニメーションの画像は「発色弾」というフリーソフトで作成しました。
URL　http://diry.net/mia

時代を先取り！ 目指せ最先端プログラマー！

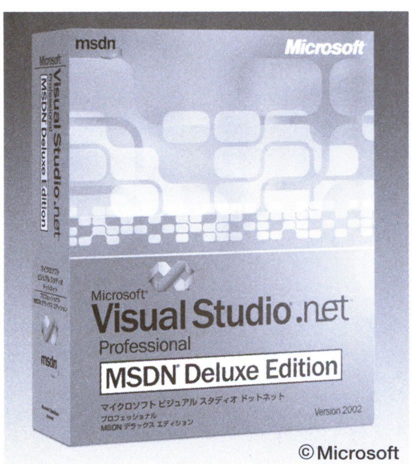
.NET（ドットネット）プログラミング入門

第3回 C#言語（後編）

廣瀬 貴一

Visual Studio .NET

この.NETプログラミング入門では，Microsoft が提唱する .NET プラットフォームのアプリケーションを，C#言語を用いて開発する方法を解説しています。.NET アプリケーションを開発するには，Microsoft の Visual Studio .NET を用いる方法が最も一般的です。グラフィカルな開発環境のため，直感的で高い生産効率を期待することができるでしょう。



《写真1》Visual Studio .NET Professional version 2002 MSDN Deluxe Edition

Visual Studio .NET は，**写真1**の Professional の場合，メーカー指定小売価格で158,000円と，高価です。Visual C#.NET Standard というものもあり，この場合は C#言語しかサポートされませんが，価格も12,800円で，初心者，入門者に適した総合開発環境も用意されています。しかし，わざわざお金を出さなくても，Microsoft は.NET Framework SDK と呼ばれる，コンパイラやリンカ，MSIL用逆アセンブラなど，基本的なツールを備えた開発キッドを提供しています。

.NET Framework SDK は，Microsoft のサイト http://www.microsoft.com/japan/msdn/netframework/ から無償でダウンロードすることができます。ただし，.NET Framework SDK は総合開発環境ではないため，コンパイルはコマンドラインから行う必要があります。

構造体

構造体は，クラスとほとんど同じ機能を提供します。フィールド，メソッド，プロパティ，インデクサなど，これらの宣言はクラスと同様に行えます。ただし，構造体のオブジェクトは値型として扱われる点でクラスと異なります。構造体は，暗黙的にSystem.Object クラスを継承するため，基底クラスを指定することはできず，継承をサポートしていないため，メンバに protected および protected internal修飾子を宣言することはできません。また，構造体は暗黙的にシールドであり，構造体，およびそのメンバを抽象化，仮想化することはできないという点でも，クラスと異なります。構造体を宣言するには，classキーワードの代わりにstruct キーワードを用います。

[attributes] [modifiers] struct identifier [:interfaces] body [;]
attributes:属性群
modifiers:構造体修飾子郡
identifier:識別子

《リスト１》構造体変数のメソッドへの値渡し

```
using System;
struct A { public string str; }

class Test {
		public static void Main() {
				A obj = new A();
				obj.str = "Default string";
				WriteString(obj);

				Console.WriteLine(obj.str);
		}
		public static void WriteString(A obj) {
				obj.str = "Update string";
		}
}
```

interfaces:構造体インターフェイス
body:構造体の本文

構造体は値型であり，構造体の変数は直接，構造体のデータにアクセスします。クラスの変数は参照型なので，異なるクラスの変数が同じインスタンスを参照することがありましたが，構造体の場合はそのようなことはなく，代入式やデータの受け渡しでは，構造体のコピーが渡されます。

リスト1を実行すると "Default string" という文字列が表示されます。プログラムは，最初にA構造体のオブジェクトを作成し，strフィールドを初期化しています。WriteLine() メソッドでこれを出力する前に WriteString() メソッドに構造体の変数を渡し，その内部でstr フィールドの文字列を上書きしていますが，なぜ，文字列が変化していないのでしょうか？

実は，構造体は値型なので，WriteString()に渡しているのはオブジェクトのコピーなのです。そのため，WriteString() メソッドは構造体のコピーを受け取り，メソッド終了時にそれを破棄します。Main() メソッド内の obj 変数は，WriteString() にコピーを渡しただけなので，結果として変化はしていません。

リスト1の構造体の宣言**struct A** の文を **class A** に書き換えてコンパイルし，実行してみれば，クラスと構造体，値型と参照型の違いが良くわかるでしょう。int型など定義ずみの数値型の変数も値型でしたが，実は，これらも構造体なのです。例えば，intはSystem.Int32 構造体の別名にすぎません。実際に変数宣言時の int 型を System.Int32 と書き換えても，プログラムはまったく同じ動作をすることでしょう。

《リスト2》構造体のボクシング変換

```
using System;

class Test {
        public static void Main() {
                int i = 10;
                Object obj = i;
                Console.WriteLine(obj);
        }
}
```

ボクシングとアンボクシング

構造体は唯一 System.Object クラス（定義ずみの別名で object 型）を継承しています。そのため，構造体もまた System.Objectにキャストすることで，汎用的な型に変換できます。しかし，System.Object クラスは参照型であり，値型の構造体とは違います。構造体のobject型へのキャストは，新たにメモリを割り当て，そこに構造体をコピーし，そのメモリ領域の参照を返すことによって，これを実現しているのです。構造体を参照型に変換することを**ボクシング変換**と呼びます。

リスト2は，int型の，すなわち System.Int32 構造体の変数 i を作成し，これを System.Object型にボクシング変換しています。逆に，ボクシングされた参照型を元の値型に変換することをアンボクシング変換と呼びます。アンボクシングは，参照が指すメモリ領域を値型変数にコピーする処理となります。ただし，アンボクシングはダウンキャストになるため，必ず明示的にキャストしなければなりません。

リスト3は，Point 構造体を System.Object 型にボクシングしていますが，その後，ボクシングした構造体のメンバにアクセスするために，WriteLine() のパラメータ部分でアンボクシング変換を行っています。予想どおり，このプログラムはPoint 構造体の x と y フィールドの値を出力します。

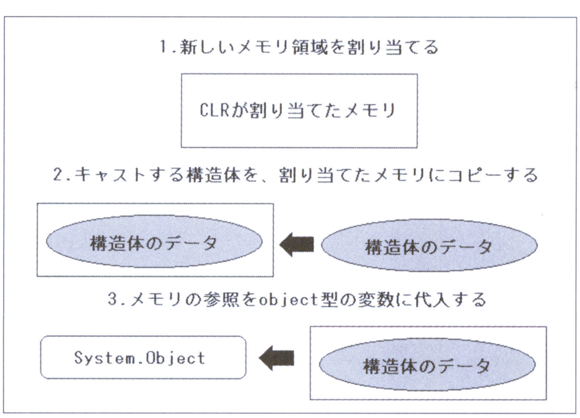
配列

C#言語における配列とは，System.Array型を継承した配列型を指し，すべての配列型は，配列宣言時に System.Array抽象クラスを暗黙的に継承します。配列の宣言は，C/C++言語などと異なり次元指定子を型の後に指定します。

type varName[]; //これは間違い
type[] varName; //これが正しい

C#コンパイラは，フィールドやローカル変数宣言で，型の後に [ ] があれば，配列型の宣言であると理解します。

C#言語では，配列もひとつの「型」であると考えられます。配列型宣言は，配列型の変数を宣言しただけで，配列のインスタンスは生成されていません。実際に何らかのデータを配列の要素に格納するには，配列のインスタンスを作成します。配列のインスタンスを生成するには，

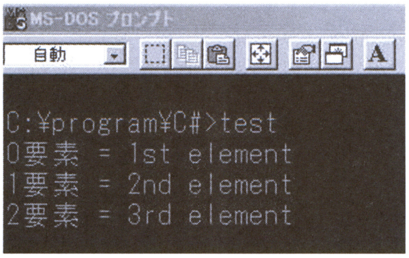


《第1図》ボクシング変換

《リスト3》構造体のアンボクシング変換

```
using System;
struct Point {
        public int x , y;
        public Point(int x , int y) {
                this.x = x;
                this.y = y;
        }
}

class Test {
        public static void Main() {
                Object obj = new Point(10 , 100);
                Console.WriteLine("x = " +
((Point)obj).x);
                Console.WriteLine("y = " +
((Point)obj).y);
        }
}
```



《写真2》リスト4の実行結果

new演算子を用いた配列生成式を用いる方法と，{ }で各要素の初期値を指定した配列初期化子を用いる2通りの方法があります。

リスト4のプログラムは，最初にstring 型の配列型を宣言し，newを用いてインスタンスを生成しています。生成した配列の要素に文字列を設定し，WriteArray() メソッドに参照を渡しています。 WriteArray()メソッドは，System.Array 型の参照としてこれを受け取ります。

すべての配列はSystem.Array抽象クラスを継承しているため，暗黙的にキャストすることができます。配列型は，必ず要素の数を自己管理し，不正なアクセスがないか常に監視しています。もし，不正な要素にアクセスした場合は，例外を発生させる仕組みになっているのです。配列の要素数はLength プロパティで取得することができ，要素はGetValue() メソッドで得ることができます。

ここで重要なのはSystem.Array型は**配列型ではない**ということです。System.Array型の変数に対して配列要素アクセスはできません。また，配列のインスタンスの生成は，**リスト5**のように配列初期化子を用いることもできます。**リスト5**の実行結果は**リスト4**と同じです。

《リスト4》配列型の宣言とインスタンスの生成

```
using System;
class Test {
        public static void Main() {
                string[] strAry = new string[3];
                strAry[0] = "1st element";
                strAry[1] = "2nd element";
                strAry[2] = "3rd element";
                WriteArray(strAry);
        }
        public static void WriteArray(Array ary) {
                for(int i = 0 ; i < ary.Length ; i++)
                        Console.WriteLine(i + "要素 = " + ary.GetValue(i));
        }
}
```

《リスト5》配列初期化子によるインスタンスの生成

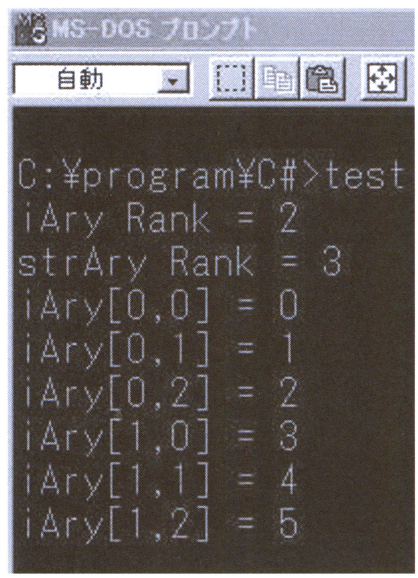
```
using System;
class Test {
        public static void Main() {
                string[] strAry = {
                        "1st element", "2nd element","3rd element"
                };
                WriteArray(strAry);
        }
        public static void WriteArray(Array ary) {
                for(int i = 0 ; i < ary.Length ; i++)
                        Console.WriteLine(i + "要素 = " + ary.GetValue(i));
        }
}
```

多次元配列とジャグ配列

C#言語で多次元配列を宣言する場合，次元指定子 [ ] 内にカンマ「,」で次元を区切ることで指定します。この時のカンマを次元分割子と呼び，次元分割子が0個の状態であれば配列は1次元なので，次元数は次元分割子数+1 となります。配列型の次元数のことをランクと呼び，System.Array クラスの Rankプロパティで，その配列のランクを取得できます。

リスト6のプログラムは，多次元配列型のiAryとstrAry変数の次元数を表示し，さらに iAry の各次元のすべての要素を表示します。多次元配列の特定の次元の要素数を得るには，System. Arrayクラスで定義されている Get Length()メソッドを用います。このメソッドは，指定した次元の要素数を返します。

C#言語の配列は「配列の配列」を作成することができます。配列の配列とは，配列の要素に他の配列の参照を持たせることによって，配列の親子関係を容易に表現することが可能であり，XMLのような木構造をもつ情報の解析などには，特に役に立つと考えられます。「配列の配列」のことを**ジャグ配列**とも呼び，配列型宣言の次元指定子を複数指定することで宣言できます。



《写真3》リスト6の実行結果

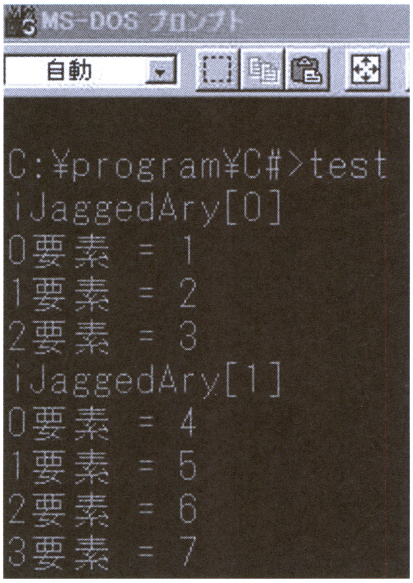
《リスト6》多次元配列の宣言と次元数の取得

```
using System;

class Test {
        public static void Main() {
                int[ , ] iAry = {{0 , 1 , 2} , {3 , 4 , 5}};
                string[ , , ] strAry = new string[1 , 1 , 2];

                Console.WriteLine("iAry Rank = " + iAry.Rank);
                Console.WriteLine("strAry Rank = " + strAry.Rank);

                for(int i = 0 ; i < iAry.GetLength(0) ; i++)
                        for(int j = 0 ; j < iAry.GetLength(1) ; j++) {
                                Console.Write("iAry[" + i + "," + j + "] = ");
                                Console.WriteLine(iAry[i,j]);
                        }
        }
}
```



《写真4》リスト7の実行結果

例えば，T[ ][ ] とすればT型の配列の配列を，T[ ][ ][ ] とすれば配列の配列の配列を宣言します。

リスト7の iJaggedAry は配列の配列です。このジャグ配列 iJaggedAry[0] は子配列の iAry1 の参照であり，iJaggedAry[1] は子配列 iAry2 の参照をもっていると考えることができます。例えばiJaggedAry[1][3]と指定すれば，それはiAry2[3] を参照するということです。

デリゲート

デリゲートは，C/C++言語の「関数のポインタ」に対応するものです。ただし，デリゲートはオブジェクト指向を基盤に作られていて，オブジェクトのインスタンスとメソッドの両方の情報を含みます。デリゲートを使うには，まずデリゲート自体を宣言しなければなりません。デリゲートの宣言には**delegate** キーワードを用いて，変数を宣言する時と同じように宣言することができます。

[attributes] [modifiers] delegate result-type identifier ([formal-parameters]);
attributes:属性群
modifiers:デリゲート修飾子群
result-type:結果型
identifier:識別子
formal-parameters:仮パラメータリスト

《リスト7》ジャグ配列

```
using System;
class Test {
        public static void Main() {
                int[][] iJaggedAry = new int[2][];
                int[] iAry1 = {1 , 2 , 3} , iAry2 = {4 , 5 , 6 , 7};
                iJaggedAry[0] = iAry1;
                iJaggedAry[1] = iAry2;

                Console.WriteLine("iJaggedAry[0]");
                WriteArray(iJaggedAry[0]);
                Console.WriteLine("iJaggedAry[1]");
                WriteArray(iJaggedAry[1]);
        }
        public static void WriteArray(Array ary) {
                for(int i = 0 ; i < ary.Length ; i++)
                        Console.WriteLine(i + "要素 = " + ary.GetValue(i));
        }
}
```

宣言したデリゲート型は，常にSystem.Delegate クラスを暗黙的に継承しています。デリゲート型は常にsealed であり，デリゲート型からさらに新しい型を派生させることはできません。デリゲートのインスタンスを生成するには，デリゲート専用の特殊な構文を用います。

new delegate-type (expression);
delegate-type:デリゲート型
expression:メソッドグループ，またはデリゲート型の値

expressionにデリゲート型の値を指定した場合，指定したインスタンスを単純にコピーします。メソッドグループは，コールバックするメソッドの名前だけを指定します。expressionがメソッドグループであれば，デリゲート型と同じシグネチャをもつメソッドが選択されます。指定したメソッドがインスタンスメソッドであれば，デリゲートはインスタンスの情報もカプセル化します。この機能を利用すれば，インスタンスに関係なく，何らかのメソッドを匿名で呼び出すことができるようになります。これは，イベント発生時に呼び出すべきメソッドの登録や，実行時に呼び出すべきメソッドを決定する，プラグイン可能なシステムの実現に使うことができます。

デリゲートを呼び出すには，デリゲートが参照するメソッドと同じパラメータで呼び出し，メソッドの戻り値と同じ値を返します。

リスト8は，単純なデリゲートによるメソッドの参照プログラムです。デリゲート型の Callback は，void M(string str)という型のメソッドの参照を含むことができ，リスト8では，これに Test クラスの Write() メソッドの参照を渡しています。この時，デリゲート型の変数methodは，インスタンスの情報もカプセル化するため，デリゲートからメソッドを呼び出すとき，暗黙的にインスタンスも参照することになります。

もちろん，デリゲートはインスタンスにこだわらないため，シグネチャさえ同じであれば他のインスタンスメソッドでも，静的メソッドでもかまいません。

また，デリゲートは加算演算子 + を用いることで複数のデリゲートを組み合わせることができます。これを**マルチキャストデリゲート**と呼び，動的にコールバックするメソッドを管理する手段として期待できます。登録したデリゲートは，減算演算子 - を使って解除することができます。

《リスト8》デリゲートによるメソッドの呼び出し

```
using System;
class Test {
        public delegate void Callback(string str);
        public static void Main() {
                Test obj = new Test();
                Callback method = new Callback(obj.Write);
                method("Kitten on your lap");
        }
        public void Write(string str) {
                Console.WriteLine(str);
        }
}
```

マルチキャストデリゲートを参照した場合，登録順に参照するメソッドが呼び出されます。ただし，マルチキャストデリゲートの戻り値は必ずvoid型でなければならず，out修飾子を使った出力パラメータを指定することもできません。

リスト9では，まずデリゲート型変数methodに，Write1()メソッドを参照するデリゲートを代入します。その後，複合代入演算子+=で，さらにWrite2()メソッドを参照するデリゲートを加算します。最後にmethodを呼び出しているので，登録されている二つのメソッドが順に参照されるでしょう。

サンプルプログラム

Microsoft .NETはXML Webサービスと連帯したプラットフォームです。.NETクラスライブラリはSystem.Xml名前空間に，XML文書を解析，操作するための洗練されたクラス群を用意しています。今回のサンプルプログラムは，C#言語を用いてXML文書を解析する簡単なプログラムを紹介します。各クラスについて，詳しくはドキュメントの.NET Framework Class Libraryを参照してください。

リスト10は，コマンドライン引数で指定されたXML文書を読み込み，開始タグと終了タグを標準出力に列挙するというプログラムです。簡単のため，リスト10のプログラムは例外処理を省略しています。本来ならば，入力されるXML文書は仕様に基いた正しい文書とは限らないため，非整形式XML文書のための例外処理も行うべきです。

System.Xml.XmlDocumentクラスは，XML文書そのものを表現するクラスです。このクラスはSystem.Xml.XmlNodeクラスから派生したクラスで，Load()メソッドで外部のXMLファイルを読みこんだり，LoadXml()メソッドで文字列からXML文書を作成したりすることができます。System.Xml.XmlNodeReaderクラスは，System.Xml.XmlNodeクラスから各ノードを解析し，ノードの型や名前などを取得することができます。

《リスト9》マルチキャストデリゲート

```
using System;
class Test {
        public delegate void Callback();
        public static Callback method;
        public static void Main() {
                method = new Callback(Write1);
                method += new Callback(Write2);
                method();
        }
        public static void Write1() {
                Console.WriteLine("This is Write1 method");
        }
        public static void Write2() {
                Console.WriteLine("This is Write2 method");
        }
}
```

《リスト10》XML文書の要素列挙プログラム

```
using System;
using System.Xml;
class Test {
        public static void Main(string[] args) {
                XmlDocument doc = new XmlDocument();
                doc.Load(args[0]);
                XmlNodeReader reader = new XmlNodeReader(doc);

                while(reader.Read()) {
                        switch (reader.NodeType) {
                        case XmlNodeType.Element:
                                Console.WriteLine("<" + reader.Name + ">");
                                break;
                        case XmlNodeType.EndElement:
                                Console.WriteLine("</" + reader.Name + ">");
                                break;
                        }
                }
        }
}
```

コラム　中学生に無料パソコン講習を実施　「IT教育運動」

パソコン検定委員会事務局（P検事務局）では，全国の中学生425万人を対象にした，無料のIT講習を4月から実施しています。

講習内容はパソコンに触れることの少ない初心者を対象としたもので，電源のオン・オフ，キーボードやマウスの操作方法，電子メールの実践や，情報モラルについて学んでいきます。講習の終了後，終了証が手渡されます。

主　催　パソコン検定委員会事務局
　　　　PASS認定校（ボランティア）
後　援　日本教育工学振興会
　　　　全日本中学校技術・家庭科研究会

受講対象　パソコンに触れる機会の少ない中学生（中1～中3生）

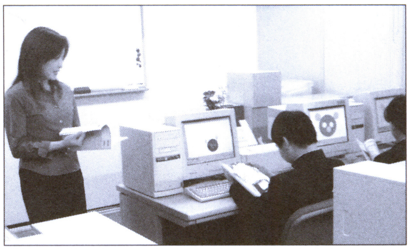
会　場　PASS認定校の会場
期　間　2002年4月～2002年12月
時　間　1人あたり3時間（分割実施可）
講　師　PASS認定校のインストラクター
カリキュラム　日本語入力とパソコンの基本操作，インターネット，電子メール，情報モラルなど
定　員　各教室提供のPC台数まで

◆講習の申し込み方法

(1)すでに各中学校に配布してある無料講習チケットに，必要事項を記入（先生の押印が必要）
(2)生徒本人または保護者が直接会場に電話予約する（講習の日時，場所を確認する）
(3)テキストを購入（講習は無料だが，テキスト代300円は有料）
(4)講習実施（チケット，テキスト，学生証を持参）

◆問い合わせ先

パソコン検定委員会事務局　03-5964-2525
P検ホームページ　http://www.pken.com/



▲全国実施に先駆けて行われた東京都渋谷区の会場には，渋谷区立原宿外苑中学校の生徒など8名が集まった

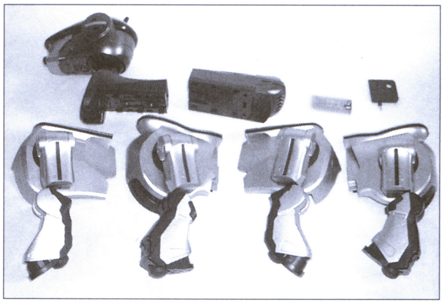
21世紀の遊びはコレ！ ロボ道場

AIBOをERS-220に変身！

220トランスフォームキット

取材／文　松原拓也

『220トランスフォームキット（ERS-220E1）』は，『ERS-210』の各ユニットを『ERS-220』に組み替えて，その変身を楽しむというキットです。　ERS-210というと「子ライオン型」のデザインが特徴的なAIBO。一方，ERS-220はメカニックなデザインのAIBOです。この2種類のAIBOは「コアユニット」が共通仕様になっています。そこで，220トランスフォームキットでは，コアユニット以外を交換，さらにコアユニットのソフトウェアを更新して，外見から機能までそっくりに変身させることができます。



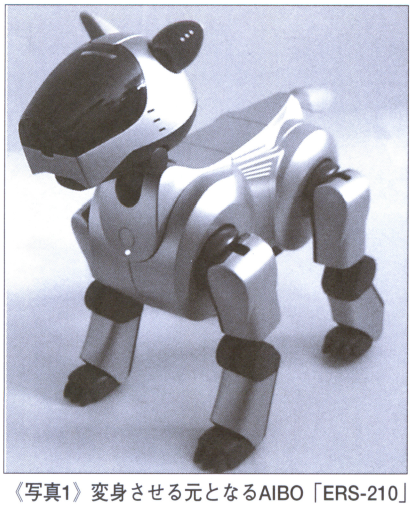
「220トランスフォームキット」
ERS-220E1
価格120,000円（税別）／発売中
*AIBO ERS-210用のアクセサリーです

セット内容

「220トランスフォームキット」のセット内容です。

・ヘッドユニット／頭部のパーツです。
・テールユニット／しっぽの部分のパーツです。
・レッグユニット／脚部のパーツです。前足2本と，後足2本の計4本です。
・バージョンアップ･メモリースティック／コアユニットをバージョンアップするためのメモリースティックです。
・リリースピン／ユニットを取り外す（リリース）ための工具です。
・マニュアル類／説明書です。

コアユニットやバッテリ，ACアダプター，ボールなどは，ERS-210のものをそのまま利用するために付属しません。



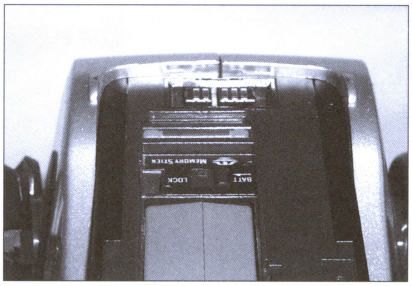
《写真1》変身させる元となるAIBO「ERS-210」

ERS-210の用意

では，トランスフォーム開始です。このキットには，変身させる元となるERS-210が必要なので，用意しておきましょう（写真1）。



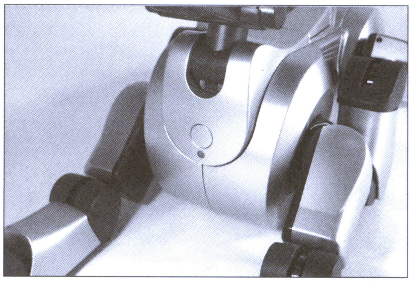
《写真2》バージョンアップメモリースティック



《写真3》コアユニットにメモリースティックを挿入



《写真4》コアユニットをバージョンアップ中



《写真5》バージョンアップ完了

コアユニットのバージョンアップ

ERS-210とERS-220とでは，ユニットの仕様に若干の違いがあります。そこで，まずERS-210に搭載されているコアユニットのソフトウェアをバージョンアップさせます（ちなみに，コアユニットをバージョンアップせずに220トランスフォームキットを取り付けると，ERS-220として正常に動作しない場合があるので注意）。

ERS-210に，バージョンアップ・メモリースティック（**写真2**）と，バッテリを挿入してから電源を起動します（**写真3**）。

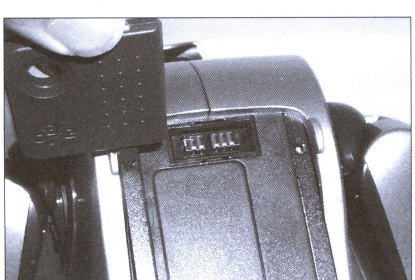
すると，メモリースティックが読み込まれ，自動的にコアユニット内のソフトウェアがバージョンアップされます。バージョンアップには1分くらいかかります（**写真4**）。バージョンアップが終了すると，胸のLEDが消灯して知らせてくれます（**写真5**）。

ユニットの取り外し

ERS-210の各ユニットを取り外します。まず，AIBOを活動停止の状態にしてから，裏返します。そして，付属のリリースピン（**写真6**）を使って，リリース用穴を押しながら，ユニットを取り外します（**写真7**）。



《写真6》リリースピン



《写真7》リリースピンを差し込んでユニットを取り外す

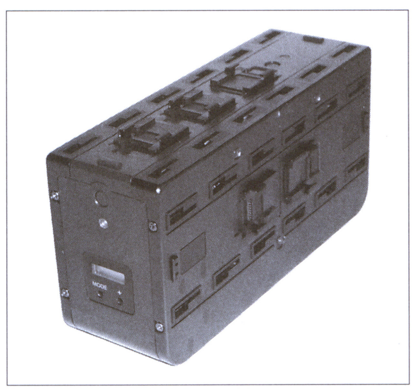
取り外していくと，最後にコアユニットが残ります（**写真8**）。ここで，ソフトウェアのバージョンが書かれたシールを貼っておきます（**写真9**）。

ユニット取り付け

コアユニットに，220トランスフォームキットのユニットを取り付けます（**写真10，11**）。ユニットとコアユニットには，接続用の溝がありますので，そこに差し込んでスライドさせると，ロックされます（**写真12，13**）。

ユニットの取り付けには特に順番がありませんが，後足からの方が入れやすいようです。

これで，組み立て完了です。見事，ERS-220に変身させることができました（**写真14**）。

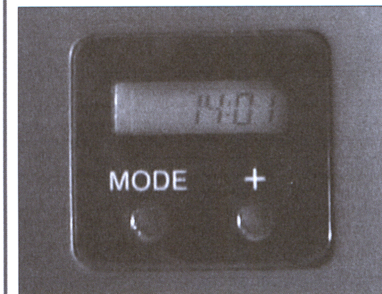


《写真8》AIBOの心臓部「コアユニット」



《写真9》バージョンアップしたことがわかるように，シールを貼る

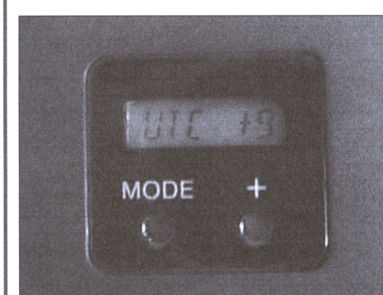
コアユニットには各設定をするための液晶パネルとスイッチがあります。



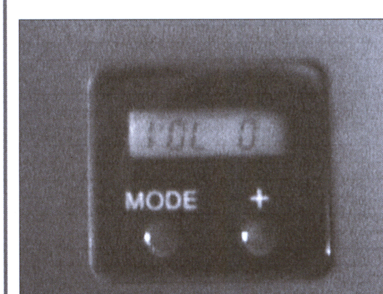
▲時刻の設定



▲日付の設定



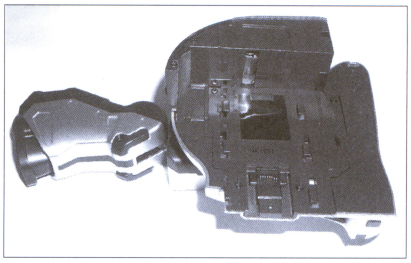
▲標準時との時差を設定します。日本時間は+9



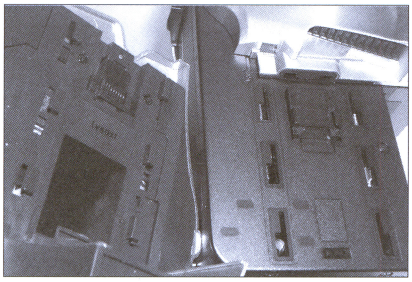
▲音量の設定。0だと，音が鳴らない



▲バッテリの残量を確認



《写真10》レッグユニット（後足）

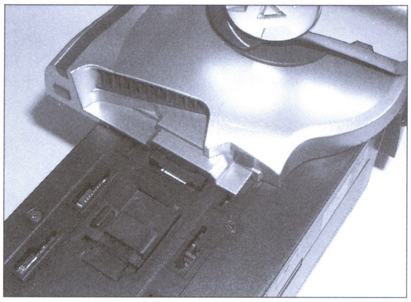


《写真11》（右）コアユニットと（左）レッグユニットの溝を合わせて取り付ける

動作チェック

では，きちんとAIBOが変身したか確認してみましょう。ためしにメモリースティックを入れずに電源を入れてみました。起動時の音がバージョンアップ前と変わっています。動作中は，ERS-210と動きが若干違うようです（**写真15**）。

他にもっとハッキリ動きを確かめたい場合は，『AIBOエクスプローラー（ERF-220AW02J）』『AIBOライフ2（ERF-220AW01J）』など，ERS-220対応のAIBO-wareを実行させてみるといいでしょう。



《写真12》ユニットを溝に合わせた様子

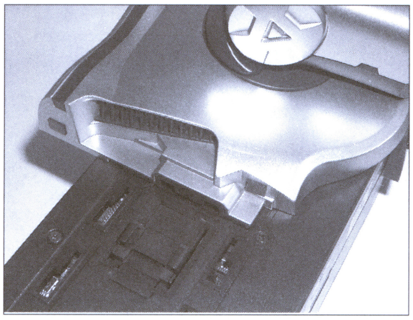
元に戻すには

これで，ERS-220に変身できたわけですが，逆に，元のERS-210に戻したい場合はどうしたらいいでしょう。実は，コアユニットは，自動的に接続されているユニットの種類を判別しますので，ERS-210のユニットに組みなおすだけで，元通りERS-210対応のAIBO-wareを使うことができます。

最後に

AIBOを変身させて楽しむ『220トランスフォームキット』。この他，ERS-220に対応したAIBO-wareとしては，AIBOを操縦して遊ぶ『AIBOナビゲーター2』や，AIBOをプログラミングして遊ぶ『AIBOマスタースタジオ』などがあります。

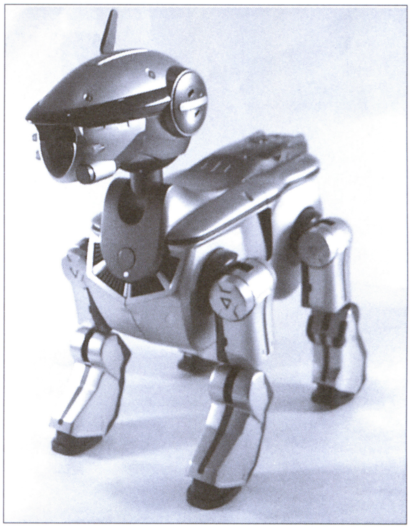
ロボットペットにホビーに，と幅広く拡大するAIBOに今後も注目です。



《写真13》ユニットをスライドさせると，カチッという音がしてロックされる



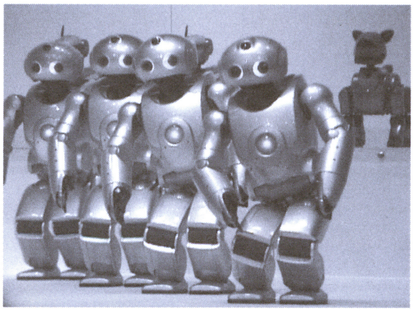
《写真14》見事，ERS-220に変身完了



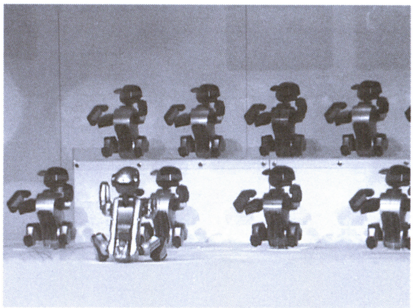
《写真15》ERS-220として，起動させた時の様子。動きの性格がERS-210と違う

ROBODEX2002でAIBOがSDR-4Xと共演　http://www.robodex.org/

3月28～31日の4日間，パシフィコ横浜でパーソナルロボットの博覧会「ROBODEX2002」が開催されました。ソニーブースの一角に設けられた「ソニードリームシアター」では，先に発表された小型二足歩行エンターテインメントロボット『SDR-4X』と「AIBO」のデモンストレーションが行われ，大評判となっていました。



▲SDR-4Xの後ろでステップを踏むAIBO



▲音楽に合わせて踊るAIBO

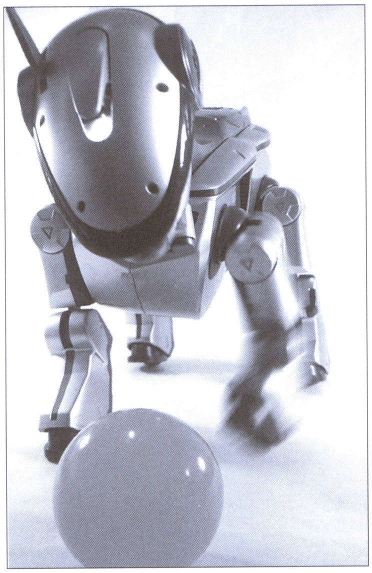
自律型AIBO-ware
「AIBOエクスプローラー(ERS-210/220用)」

「AIBOエクスプローラー」は，よりロボットらしい動きとサウンドで，アクティブに探索して回る自律型のAIBO-wareです。

音声でコミュニケーション

たとえば，ユーザーが「ぼーるさーち」と話しかけると，ボールを探し始めます(**写真16**)。「あくしゅ」と話しかけると，AIBOがお手をしてくれます（**写真17**）。「よしよし」と誉めたり，「だめだめ」と叱ったりすることができます。

認識する音声の数は約75です(**第1表**)。しばらく使ってみると，マニュアルに載っていない，思いがけない言葉にも反応してくれました(**写真18**)。他にもいろいろな発見があるかもしれません。



《写真16》ボールで遊ぶAIBO

AIBOが2体以上いれば，AIBO間でコミュニケーションすることも可能です。他のAIBOからの話しかけに反応し，AIBO同士の会話もはずみます。

他に，AIBOに名前をつけて登録したり，AIBOにユーザーの名前を覚えてもらうことができます。

ブーストモード

AIBO同士やユーザーとのコミュニケーションが上手くいくと，AIBOの感情が高まって，「ブーストモード」に入ることができます。このブーストモードでは，AIBOがハイテンションでアクロバティックな動きを見せてくれるそうです（**写真19**）。



《写真18》いろいろな芸を披露するAIBO

その他の機能

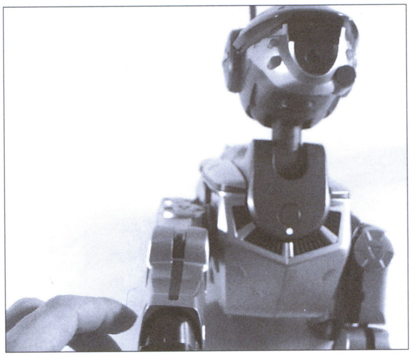
その他，音声認識，写真撮影，名前登録・認識，モーション登録，AIBO間コミュニケーション，サウンドインタラクション機能（音遊び）等があります。音を使った遊びでは，ものまねをはじめ，AIBOメロディーやAIBOハミングなどの機能があります。

ERS-210/220両対応

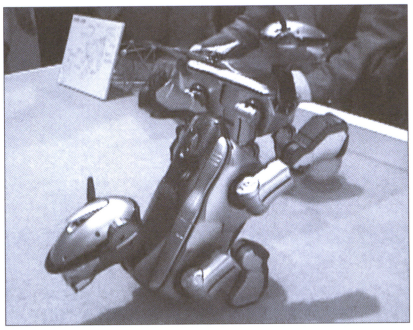
「AIBOエクスプローラー」は，ERS-210とERS-220の両方に対応していますが，その動作には若干の違いがあります。

ERS-210は反応に合わせて耳や口，しっぽが動きます（**写真20**）。これらの可動するアクションは，ERS-220にはありませんので，しっぽや顔にあるのLEDの点灯で表現します。あとERS-220には，リトラクタブルヘッドライトがあり，興奮した時などにポップアップして点灯します。

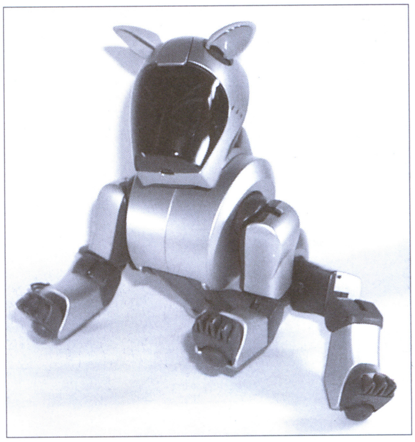
『AIBO エクスプローラー
(ERS-210/220用)』
ERF-220AW02J
価格9,000円（税別）



《写真17》お手をしているAIBO



《写真19》逆立ちをするAIBO



《写真20》ERS-210で実行した時の様子

《第1表》AIBOが認識する言葉の例（一部です）

言葉	AIBOの反応
こっちこっち	AIBOが，呼んだ方にやってきます。方向が正確にわからない場合もあります。
ものまね／ものまねしよう／おはなししよう／あそぼう	AIBOの言葉で，ものまねを始めます。
あるいて／たんさく	AIBOがお部屋探索に出かけます。
あいぼつうしん／おともだち	AIBO同士をコミュニケーションさせるときに話しかけます。
めかにかるあくしょん	メカっぽい動きをします。
しゃしんとって	AIBOから今見えている風景の写真を撮ってくれます。

歴史ゲームはもう難しくない!?
ゲームで楽しく「歴史」を学ぶ!!

歴史の授業といえば，あまり人気がないイメージがありますが，流れのある一本のストーリーとして捉えると面白いものです。歴史を好きになる人にはさまざまな契機があるでしょうが，ゲームがきっかけになった人も多いといわれています。

今月紹介する『三國志孔明伝』は日本でも古くから人気のある小説『三国志演義（以下，演義と略す）』の世界を舞台にしたRPGスタイルのリコエイション・ゲームです。（文／源康史）



三國志孔明伝

■コーエー　■リコエイションゲーム　■Windows95/98/Me　■1，980円　■発売中

遊びやすさを追求した三国志

『孔明伝』はシナリオを追いながらキャラクターを成長させていくRPGのようなシステムです。三国志がどんな話なのか，てっとり早く楽しむことができるので，SLGが苦手という人でも心配いりません。登場武将200人のうち，「孔明」の視点でストーリーが展開されていきますが，肝心の歴史にとって大切な前後のストーリーについては，プロローグとして事細かに紹介されますので，話の流れがよくわかるようになっています。

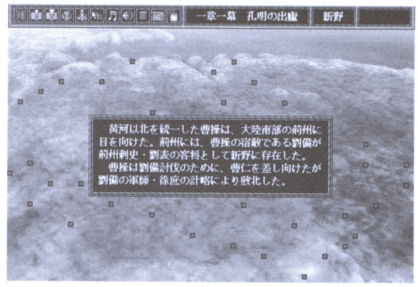
原作の『演義』は，公認の歴史書として認められている『正史』と民間伝承，講談のタネ本などを参考に，明初の羅貫中（らかんちゅう）が著したといわれています。

『演義』の場合，半分近くはフィクションですから，劉備（りゅうび）軍をひいきしたり、妖術やまだ発明されていない火薬などが登場しますが，流れは正史と変わりませんので，歴史を学ぶきっかけとしては打ってつけです。

『孔明伝』の舞台は2世紀末から3世紀の中国。今日でも漢字や漢民族と名を残すほど歴史上重要な漢王朝も次第に衰え，農民を中心にした大規模な宗教叛乱「黄巾（こうきん）の乱」が起き，皇帝の権威が地に落ちていきます。この乱がきっかけで，各地の群雄が実力をもって統治する乱世の時代に突入していきます。その動乱の中，ライバルを次々と破り，勢力を伸ばしていったのが曹操で，後漢皇帝（献帝）を擁し，事実上中国大陸の覇者となります。

孔明は181年（黄巾の乱の3年前）に徐州瑯邪（ろうや）で生まれます。三国志初期の群雄に比べると子の世代に当たります。

諸葛（しょかつ）が姓で，名は亮（りょう），一般的によく知られる「孔明」は字（あざな）といい，成人後につけられる「通り名」です。その後，曹操（そうそう）の徐州侵攻により荊州（けいしゅう）に移住し，27歳まで半ば隠居するような生活で，ひたすら学問に励んでいました。そんな時，荊州に来ていた劉備に誘われます。劉備は孔明を軍師にしたのを契機に，蜀（しょく）（益州）を領するようになり，曹操の魏（ぎ），孫権の呉（ご）に対抗する群雄へと成長していきます。



▲イベント前には必ず，背景説明が入る。孔明が登場するのは三国志中期になる。



▲少年時代の孔明が徐州を追われていく場面。曹操に仕えなかった一因かも??



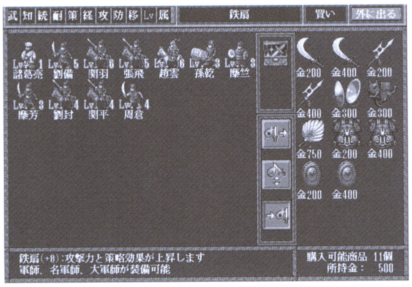
▲劉備の軍師だった徐庶が，孔明を劉備に推薦する。だが……



▲孔明は自分の目でこれと決めた主君以外に仕える気はなく，志も大きい。

ゲームの流れは簡単でスムーズ

軍備増強



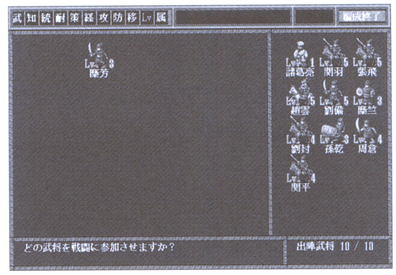
▲部隊を強化しよう。お金があれば武器屋で武力・防御力をアップできる。道具屋では回復の薬などを買おう。

作戦の決定



◀難易度が変わる!? 戦の前に孔明が幾つか作戦を出してくれる。選択によって命運も変わる。

出陣



▲兵科を把握しよう!! 劉備軍が最初に使える兵科は歩兵系，弓兵系，騎兵系，軍師系，補給系，山賊系で補給系では食料部隊と軍楽隊を率いることができる。糜竺の軍楽隊で補助を優先させるか，糜芳の歩兵隊を出すか迷うところだ。

作戦を指示する
念を入れる

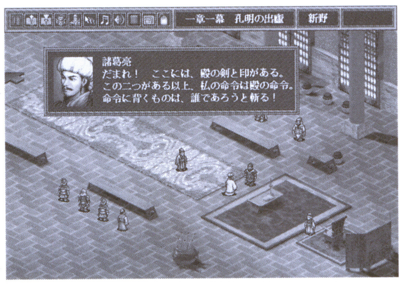
火計を行う
敵大将をねらう

ちょっと冒険



▲ちょっと冒険。慎重にいかないと関羽と張飛が命令に従わず，勝手に出撃してしまう。苦戦は必至!?

演義の通りに進める



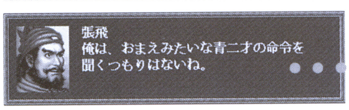
▲演義のとおり進める。関羽と張飛は伏兵，趙雲は囮部隊になり，夏侯惇を狭道に誘い込む。余裕の展開だ。

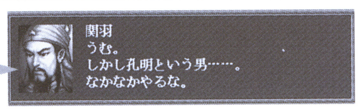
火計で敵は大混乱



◀火計で敵は大混乱!! ワナにハマった夏侯惇軍は混乱し，しばらく反撃してこないので楽勝だ!!

書生から軍師へ





◀勝利により，関羽と張飛もやっと孔明を一人前と認めてくれるようになる。

ゲームの流れはイベントと戦闘の繰り返し!

歴史SLGの敷居を高くしているのは，内政や外交や戦闘の複雑さだと思います。むろん，それらはゲームに深みをもたせる大切な要素ではあるのですが，『孔明伝』ではそれら複雑な要素は一切ありません。

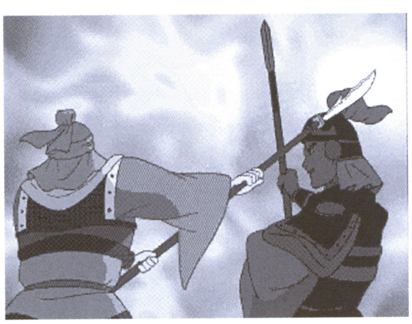
初戦では博望坡（はくぼうは）で夏侯惇（かこうとん）軍を迎え撃ちますが，初陣だけあって難易度は低めに設定してあります。『演義』での孔明は派手な勝利ばかりですが，実際に読んでみて，異なる戦略を使ってみると面白いかもしれません。戦闘では「勝利条件」をクリアするだけで先に進めます。博望坡では夏侯惇の撤退が条件になりますが，その前に雑魚をより多く倒し，経験値を稼いでおきましょう。

『演義』を読みたい人は書店を探すか，注文するなどしてみましょう。図書館にもありますよ!!

『三国志演義』上下巻　平凡社刊
『三国志』全8巻　吉川英治著　講談社

吉川英治氏の『三国志』は日本人のための三国志で，日本人にもわかりやすくアレンジされている名著です。

目的達成以外にも楽しみ満載!!



◀名のある将軍を攻撃すると，たまに一騎討ちになることがある。見逃すな!!

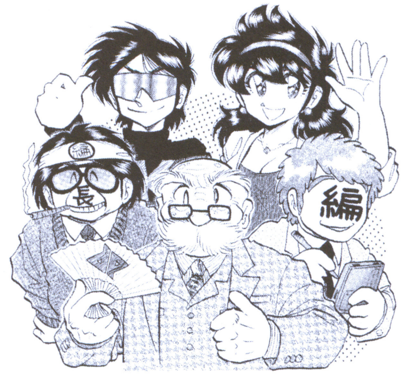


◀兵糧庫や宝物庫にはアイテムが落ちている。無理してでも取りに行こう!!

三國志孔明伝略年表

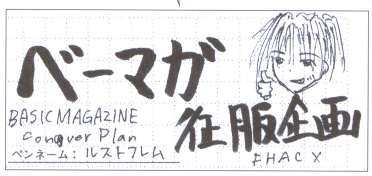
181年	諸葛孔明、徐州に生まれる
184年	黄巾の乱勃発
190年	董卓が献帝を擁する
193年	曹操徐州攻略，孔明荊州へ
200年	曹操，官渡で袁紹を破る
208年	孔明，劉備に仕える
	曹操，漢の丞相になる
	曹操，赤壁で呉に敗れる
214年	劉備，益州（蜀）を領す
216年	曹操，魏王になる
220年	曹操、洛陽に没す
	後漢滅亡，曹丕が皇帝に
221年	劉備，蜀皇帝になる
	孔明，蜀丞相になる
223年	劉備、白帝城に没す
229年	孫権，呉皇帝になる
234年	孔明，五丈原に没す

あくせすROOM



影：だんだん暑くなってきた！雨の日などは急に冷えこんで，カゼを引いてしまい仕事どころではない。昼間はムシムシしていて，仕事どころではない。
編：いいから，原稿書いてください。

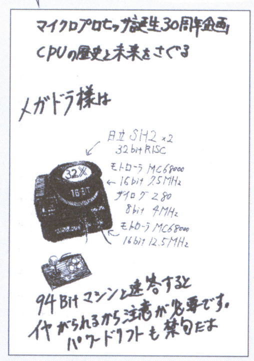
タブレット
ほしい



▲東京都練馬区・ルストフレム

編：まずは，あくせすROOMから席巻してください。
影：かわいいイラストで私のハートを席巻してくれ！

文字ネタ部門
御中



◀福島市・南斗オレガ

バブルン
さがします



◀神奈川県平塚市・ら一抹茶

編：ベーマガのロゴ入り。友達に自慢しよう！
影：早くしないと半年分しか見せられないぞ。
編：ちぎらなきゃいいじゃないですか……。

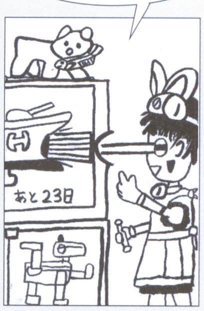
編：イヤがられるもなにも，この理屈が通るんだったら，CPUを何個もつけているサーバーマシンは，最終的に何ビットマシンになるんでしょうか。
影：OSが対応してないかぎり，あきらかに，そのまんまでしょ。
Dr.D：さりげなく描いてあるこのコントローラは……。

▼愛知県名古屋市・わいち301



つぐ美：ラクガキって，描くのも楽しいし，人の描いたのを見てても，暖かみがあるのよね～。みんなもラクガキ気分で，気軽に投稿してね！
影：パーツの設定しだいで，いろいろなアニメーションができるあたり，かなりテクニカルな攻略要素もあったりするし。

やってません？
誰か!?



◀愛知県名古屋市・最馬場可

影：「思いつき」をそのまま描き殴っていたのが，最近は再構成し，いろいろなテーマを内包するようになってきた。あと一歩で芸術の域にまで達するぞ。
編：コラージュっぽい楽しさがありますよね。

うろ覚え選手権　今月の課題「世界の果て」



◀福島市・南斗オレガ

編：「世界の果てにこんなのが見えた」とのこと。

影：お題が難しすぎたのか，抽象的すぎたのか，投稿がほとんどこなかったりして……。こういうときは，世界の断崖から落ちた船やら車やらが，うずたかく山になっているとか，記念品をもらって折り返す場所になっていて簡易トイレが1個だけあるとか…。ありそうな情景を絵にしてみるだけでも，じつは人それぞれの表現ってのがあって，それはそれで楽しめるものなんだけどなあ。次回のテーマは，「振り返る直前の背後の世界」だけれども，みんなの想像力はどこまでハジけてくれるのかな？
編：みんなが慣れてくるのを待たずに，そろそろ企画自体新しくしましょうか。
影：突然の出題にも素早く対応できるのも，投稿者に求められる資質の一つ。

あくせすROOMの参加方法　〈投稿歓迎〉

あくせすROOMでは，みなさんの投稿をお待ちしています。愛読者カードや，官製ハガキ，または同サイズの丈夫な紙などを封書でまとめるなどして，お送りください。

鉛筆や色鉛筆はきれいに印刷されません。せっかくペンで描いてあっても，鉛筆の下書きが残っているものは，やはり掲載しにくくなりますので，インクや水性ボールペンなど，はっきりと書ける道具を使いましょう。文字だけの投稿であれば，ワープロを使ってもかまいませんよ（そのまま掲載できます）。

★応募先
〒141-8715　東京都品川区東五反田1-11-15　電波新聞社マイコンBASICマガジン編集部「あくせすROOM」係

《募集物の一例》

●うろ覚え選手権
7月号の募集：あなたが振り向く直前まで，背後に存在していた世界を想像してみてください。鏡越しだと見えない「なにか」が，そこにはあるはずです。5月15日必着です。

●言い訳の達人
失敗を恐れて，行動を起こす前から弁明を考えてしまう若者に，もっと時間を大切に使ってほしいと願う。いろんな失敗を想定して，おもしろい言い訳を考えてみてください。文章，イラスト，両方で募集しています。

●世界のカスタム野郎
愛用の道具をいろいろカスタマイズ（自己改造）して使っている人の自慢話を募集します。冷蔵庫にパーツを組み込んでDOS／Vパソコンを作る！ゲームコントローラに新しく機能をつける！　ちょっとした工夫で，実用性がアップしたり，カッコ良くなったという実例を，ハガキ1枚をメドに紹介してください。

●デジカメ激写ボーイズ
デジカメを持って街に出よう！　デジカメやデジビで撮影したユニークなお笑い画像を送ってください。タイトルはアレですが，女性のかたの投稿も歓迎です。

あくせすROOMは，読者のみなさんのパワーで作るページです。このほかにも，おもしろいネタがあったら，文字でもイラストでも，思いつくままに表現してみてください。読者提案の企画として長くつづくポイントは，ほかの人がちょっとしたアレンジで相乗りできること。みんなで楽しく参加できそうなネタを常時募集しています。

6月号お楽しみプレゼント

①アップ・スキャンコンバーター・ユニット「XRGB-2plus」／1名（提供:マイコンソフト社）
②各社の携帯ストラップ／5名（提供：マイコンBASICマガジン編集部）



●応募方法：本誌とじ込み愛読者ハガキ（またはインターネット読者ハガキ）にプレゼント記入欄があります。一つだけ選んでお送りください。
●締め切り：2002年6月7日必着

《2002年4月号の当選者》①岐阜県羽島市・川田博美さん他2名　②福島市・山岸吉弘さん

Falcom® NEWS EXPRESS

日本ファルコム株式会社
〒190-0012 東京都立川市曙町1-14-13
TEL 042-527-6501
FAX 042-528-2714
スタッフ募集中
採用担当BFM係
http://www.falcom.co.jp/

遂にタイトル公開！ファルコム完全新作！

VM JAPAN
VANTAGE MASTER - MYSTIC FAR EAST

すでにご存知の方も多いと思いますが，ファルコムの次回新作は「VM JAPAN」（ブイエムジャパン）です。そう，タイトルのVMを見てもわかるように，あのヴァンテージ・マスターの最新作なのだ！といっても，単なるMAPを新しくしただけの，単なる続編ではないぞ！基本的なルールは踏襲しているものの，世界観，キャラクター，さらにはシステムもすべて一新されているので，前作をかなりやり込んだ人も，新たな気持ちでプレイできるぞ！

主要キャラ紹介第2弾！今月は敵キャラだ！

先月紹介したキャラクターはVM JAPANの召喚士。

つまり前作のマスターにあたる，いわゆるプレイヤーの分身でもある主人公たちだ。で，これから紹介するのは，その主人公たちと対戦する敵キャラです。今回は敵キャラにも設定と役割があるので，前作以上にストーリーにも深みが増しているぞ！



◀メリー・ルー
なぜか外人の女の子。実は重大な使命があるらしいのだが……。回復系の術を中心に使ってくるぞ。中でもやっかいなのは，全回復させる術をもっていること。追い詰めたと思っていても一発逆転させられる恐れがあるのだ。

▼イサミ
とにかく防御力も高い上に，攻撃力は全キャラ中最強！斬り込まれたら一発でやられることもあるので，接近戦だけは避けるべし。衣装は新撰組がモチーフ？苗字は近藤かどうかは不明。

▼シャダイ
いかにも悪者の雰囲気バリバリのこのキャラ。能力的には，さほど高くはないけど完全回復させる術と，レンジの長い攻撃の術をもっており，なかなかしぶといのだ。しかも，相手のHPにダメージを与えつつ，自分を回復させる術には注意が必要だぞ。



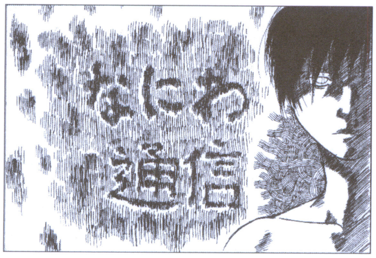




▲ガモン
こいつのやっかいなところは，攻撃力が高い上に離れたところからガンガン撃ってくるので，攻撃レンジの見極めを誤ると大変なことになるぞ。術に対する防御力も高いので，かなりやっかいな相手だ。

今回のVM JAPANでは，シナリオモードに，新たにハードモードが追加されているのだ。もちろんイージーモードもあるので，初心者には優しいんだけど，上級者には容赦なく厳しくなっているぞ。ハードモードは1面からいきなりエキスパート並の難しさ！自信がある人はぜひハードモードに挑戦しよう！

最新情報はファルコムホームページで！ http://www.falcom.co.jp/



DirectX9対応間近！
なにわ通信

タイトル冠文：大阪府・天満橋さくら
タイトル・イラスト：大阪府・ARAI
本文：暁

PRESENTED BY マイコンソフト ADDRESS● 〒530-0005大阪市北区中之島3-2-4
朝日新聞ビル6F電波新聞社大阪本社内
マイコンソフト株式会社 なにわ通信係
http://www.micomsoft.co.jp

前号に引き続き台湾特集です。

その前にちょっと訂正。前号で，台湾にはファミリーマートが多いと書きましたが，台北市内にはセブンイレブンの方が多いみたいです。日本同様，地区によってコンビニの勢力が違うようです。

電気街・八徳路

台北に行ったら一度は顔を出しておきたい場所があります。

日本には秋葉原など電気街と呼ばれる場所がありますが，台北にもやっぱりあるのです。

「八徳路(パァートゥール)」というところが，いわゆる電気街で，DOS/Vショップなどが建ち並びます。さすがに台湾はDOS/Vパーツの本場だけあって，日本に有るようなパーツはほとんどあります。

しかし，本体ケース以外のパーツは，あまり安くなく，かえって日本の方が安かったりします。マザーボード等も，ほとんどが台湾製なのに，あまり日本と値段は変わりません。

不思議なことにマイクロソフトのOSやアプリケーションは日本とは比べ物にならないほど値段が高かったですね。

電気街を歩いていて強く印象に残ったのが，ノートパソコン専門店です。ノートパソコン専門の販売店が数軒あるのですが，どの店もショールームのような店構えで，必ずたくさん椅子とテーブルが置いてあり，じっくり商談できるようになっています。

台湾では，「ノートパソコン=高級品」らしく，まだ持っている人は少ない様です。ノートパソコンは「お金持ちの買う物」，と相場が決まっているらしく，これら専門店はそういったリッチな方々を迎えるため，店内もお洒落な雰囲気にしているようです。

ちょっと入り辛かったので，結局店内には入りませんでしたけどね。

高架商場

秋葉原にはラジオデパートと呼ばれるお店がたくさん入っている建物が有りますが，台湾にも高架商場(クワンホワンシャンチャン)というところがあって，八徳路の代表的な建物になっています。

中に入ると，パソコンのパーツやら，日本でも発売されたばかりのゲームや，電子パーツのお店が軒を連ね，人混みをかき分けながら進まなければなりません。

国は違っても，この手の場所はいつも賑やかですね。

卒業旅行だと思うのですが，日本の大学生らしき人達が数人で店員と値段交渉をしている光景を目にすることができました。

台湾旅行のススメ

台北は皆さんも是非行ってみることをお薦めします。台北市内の治安は比較的良いようです。どこの国でもそうだと思いますが，荷物の盗難に気をつけておけば，トラブルは起きないと思います。

次号も引き続き台湾特集をする予定です。お楽しみに。

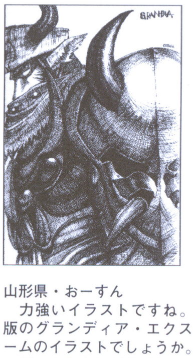


写真1：台北の秋葉原こと「八徳路(パァートゥール)」の様子。いつも人がいっぱいです。

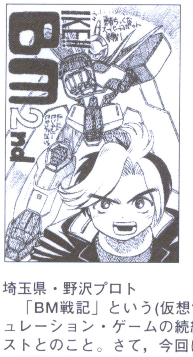


写真2：マニアが集まる高架商場。台北の電気街はここから始まった。

ありがたい，おハガキのコーナー



山形県・おーすん
力強いイラストですね。PS2版のグランディア・エクストリームのイラストでしょうか。
ゲーム内容はどうでしたか？



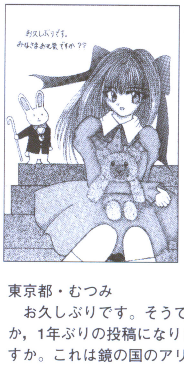
埼玉県・野沢プロト
「BM戦記」という(仮想?)シミュレーション・ゲームの続編イラストとのこと。さて，今回はどんな活躍を見せてくれるのでしょう。



大阪府・藤井朱丸様江
「一名様ご案内」という題名のイラストです。マヲさんて，のんびりとした感じのイラストが得意ですよね。



大阪府・ARAI
待ってました！待望の新シリーズが始まりましたね！とても元気な動物みたいですが，名前は何と言うのでしょう？



東京都・むつみ
お久しぶりです。そうですか，1年ぶりの投稿になりますか。これは鏡の国のアリスでしょうか？

◆ 定期購読●直接購読のしおり ◆

弊社刊行の雑誌は，ご自宅直送方式の年間定期購読をお勧め申し上げます。スピーディーで確実，そしてお得な定期購読をぜひご利用下さいませ。

また　ご希望の雑誌・書籍が，ご近所の書店にない場合は右の振替用紙にて直接ご注文下さい。

■ 年間定期購読料（消費税込料金となっております。）

電　気　店	￥16,000	マイコンBASICマガジン	￥6,120
電波新聞縮刷版（CD-ROM版）	￥96,000		

■ 年間購読の特典

①特別号も通常の定価でご購読いただけます。
②電気店は13冊分の購読ができます。

◆確実な振替口座を利用してご送金下さい◆

送金手数料が不要で，確実な郵便振替でご送金をお願い申しあげます。手続きは簡単です。右の郵便振替用紙に下記の必要事項を記入し，郵便局窓口に必要金額と共にご提出いただくだけですみます。

①購読申込書に必要事項を記入ください。なお１年間，或は２年間購読かをはっきりご記入ください。
②払込通知票，払込票に送金合計金額と住所，氏名をご記入ください。払込通知票と払込票は絶対に切りはなさないようにしてください。なお，必ずフリガナをご記入下さい。
③弊社への御意見・御希望がございましたら，通信欄へご記入下さい。
④すべて記入が終了しましたら，案内文のみ切り離して郵便局窓口に金額を添えて提出してください。

※振替によるご送金は，着本までに約２週間ほどかかりますのでお含みおきください。
お急ぎの場合は，現金書留に，右購読申込書を同封してご送金ください。

この受領証は、郵便局で機械処理をした場合は郵便振替の払込みの証拠となるものですから大切に保存してください。

ご注意

この払込書は、機械で処理しますので、口座番号及び金額を記入する際は、枠内に丁寧に記入してください。

また、下部の欄（表面及び裏面）を汚したり、本票を折り曲げたりしないでください。

（郵政事業庁）

払込取扱票

通常払込料金
加入者負担

02 東京

口座番号（右詰めにご記入ください）

0 0 1 5 0 - 3 - 5 1 9 6 1

金額 千 百 十 万 千 百 十 円 ※

加入者名 株式会社 電波新聞社

料金　　特殊取扱

通信欄

※ 定期購読誌名	購読期間	送金額（送料当社負担）
※新規 継続	年 月号より 年間	円
御希望誌・ソフト・グッズ名	冊数	送金額（送料共）
		円
		円

フリガナ
送付先住所
□□□-□□□□

産業分類	所属又は職種

フリガナ

電話	年令	性別
TEL	才	男・女

払込人住所氏名 （郵便番号　　　　）
※
（電話番号　　－　　－　　）

受付局日附印

各票の※印欄は、払込人において記載してください。

裏面の注意事項をお読みください。郵政事業庁（私製承認 東 第1005号）

切り取らないで郵便局にお出しください。

記載事項を訂正した場合は、その箇所に訂正印を押してください。

払込票兼受領証

口座番号 0 0 1 5 0 - 3 　通常払込料金加入者負担
右詰めにご記入ください
5 1 9 6 1

加入者名 株式会社 電波新聞社

金額 千 百 十 万 千 百 十 円 ※

払込人住所氏名 ※

料金（消費税込み）　円

特殊取扱

受付局日附印

切り取って郵便局へお出し下さい。

電波新聞社出版物一覧

（消費税込料金となっております）

2002. 3 現

最新エレクトロニクス用語事典	5,300円	〒380円	F-BASIC V6.3プログラミングのすべて	2,835円	〒310
半導体・ICのすべて	2,520円	〒310円	まんがぱそてんシリーズ	各	各
ビデオ技術のすべて	2,100円	〒310円	第1巻(基礎編)/第2巻(応用編)/第3巻(ネットワーク編)	1,260円	〒310
TVのすべて改訂版	1,988円	〒310円	パソコン認定試験合格対策試験問題集3･4級	2,100円	〒310
DVD&ホームシアター	1,680円	〒310円	シスアド受験スーパー記憶術	2,900円	〒340
新デジタル映像技術のすべて	2,100円	〒310円	基本情報受験スーパー記憶術	2,835円	〒340
センサーのすべて	2,000円	〒310円	初級シスアド受験マニュアル	3,780円	〒450
改訂版DVD&DVC入門基本18章	1,529円	〒310円	基本情報受験マニュアル	3,780円	〒450
増補改訂版デジタル放送がわかる本	1,600円	〒310円	デジタル1種〈技術科目〉最新テキスト	2,205円	〒340
増補改訂版・新ビデオ技術ハンドブック	2,500円	〒310円	デジタル3種受験マニュアル	2,940円	〒340
新・まんが電気学入門	1,500円	〒310円	アナログ・デジタル全資格〈法規科目〉	1,995円	〒340
新電気学入門講座	2,050円	〒310円	特殊無線技士受験マニュアル･2002年版	2,940円	〒340
新オシロ・ロジアナ入門講座	2,243円	〒310円	オールアバウト カプコン対戦格闘ゲーム1987～2000	2,100円	〒450
電子計測器ガイドブック	7,500円	〒450円	オールアバウト SNK対戦格闘ゲーム 1991～2000	2,310円	〒450
よくわかるプラズマテレビ(PDP)	1,680円	〒310円	電子パーツ物知り百科	1,890円	〒310

※上記以外にも新聞・出版物がございますので最寄りの本社、支局へお問い合わせ下さい。

東京本社	03-3445-8201(ダイヤルイン)	新潟支局	025-245-2526	京都支局	075-221-8021
大阪本社	06-6203-3361	関東総局	027-326-3206	広島支局	082-228-558ｹ
西部本社	092-431-7411	松本支局	0263-36-2266	高松支局	087-861-3111
札幌支局	011-641-5591	名古屋支局	052-261-4541	鹿児島支局	099-226-3630
仙台支局	022-227-7211	金沢支局	076-263-8661		

キリトリ線

月刊マイコンBASICマガジン⑥ 愛読者カード

★OFコーナー宛 ♥あくせすROOM宛 ■質問・感想（本誌への掲載／可・不可）

ペンネーム：

■アンケート回答記入欄■

※黒の筆記用具でわかりやすく，はっきりとご記入ください

Q1	□	Q2	□	〔メーカー名　〕				
Q3	□	Q4	□	Q5	□			
Q6	□	□	□	□	□	□	□	□
Q7	□	□	□	□	□	〔商品名　〕		
Q8	□	□	□	□	□	□	□	□
Q9	□	□	□	□	□	□	□	□
Q10	□	Q11	□	〔④の人はソフト名を記入　〕				
Q12	〔　〕	Q13	〔　〕					
Q14	□	□	□	Q15	□	□	□	

■アンケートにご協力ください。

※左の愛読者カード・記入欄に，以下の質問に該当する番号をご記入ください。複数回答の質問で，回答欄をすべて埋める必要はありません。

Q1　メインで使っているパソコンは？
①デスクトップ ②ノート ③持っていない

Q2　パソコンの種類は？（Q1で③以外の方）
①自作 ②ショップブランド ③メーカー製
※枠内にメーカー名も記入してください

Q3　メインで使っているOSは？（Q1で③以外の方）
①Windows 95 ②Windows 98 ③Windows Me ④Windows 2000 ⑤Windows NT ⑥Windows XP ⑦Macintosh ⑧UNIX ⑨Linux ⑩その他

Q4　パソコン歴は？
①1年未満 ②1～2年 ③3～5年 ④6～10年 ⑤11年以上 ⑥なし

Q5　インターネット歴は？
①1年未満 ②1～2年 ③3～5年 ④6～10年 ⑤11年以上 ⑥なし

Q6　パソコンの主な使用目的は？（複数回答可）
①プログラミング ②ゲーム ③インターネット ④メール ⑤ミュージック ⑥学習 ⑦CG ⑧その他

Q7　これから買いたいと思うコンピュータ機器は？（複数回答可）
①パソコン本体 ②プリンタ ③スキャナ ④デジタルカメラ ⑤その他
※わかれば，商品名も記入してください

Q8　持っているゲーム機は？（複数回答可）
①プレイステーション（PSone含） ②プレイステーション2 ③ドリームキャスト ④ニンテンドウ64 ⑤ゲームキューブ ⑥Xbox ⑦ゲームボーイ ⑧ゲームボーイカラー ⑨ゲームボーイアドバンス ⑩ワンダースワン ⑪ワンダースワンカラー ⑫その他

Q9　これから買いたいと思うゲーム機は？（複数回答可）
①プレイステーション（PSone含） ②プレイステーション2 ③ドリームキャスト ④ニンテンドウ64 ⑤ゲームキューブ ⑥Xbox ⑦ゲームボーイカラー ⑧ワンダースワンカラー ⑨ゲームボーイアドバンス

※裏面へつづく→

キリトリ線

■メモ欄…プログラム担当者・編集部へのご意見・ご感想など

Q10 携帯・PHSを持っていますか？
①両方持っている ②携帯 ③PHS ④購入予定 ⑤予定なし

Q11 ワープロ・ソフトは何をお使いですか？
①Word ②一太郎 ③メモ帳 ④その他

Q12 使用中のプログラミング言語は？
※枠内に一つご記入ください

Q13 これから勉強したいプログラミング言語は？
※枠内に一つご記入ください

Q14 今月号の本誌で面白かった記事を三つ，下の表からお答えください

Q15 今月号の本誌でつまらなかった記事を三つ，下の表からお答えください

ご協力ありがとうございました。

■今月号の記事一覧

キリトリ線

マイコンBASIC Magazine プログラム投稿応募カード

言語名（音源名）					
プログラム名（曲名）					
氏　名		ペンネーム		年齢	歳
住　所	〒				
TEL	※連絡可能な時間帯			応募年月日	年　月　日

キリトリ線

郵便はがき

50円切手をお貼り下さい

141-8715

（受取人）

東京都品川区東五反田1−11−15

電波新聞社

月刊 マイコンBASICマガジン 編集部

6月号 行

（フリガナ）		年齢	性別
ご氏名		歳	男・女
□□□-□□□□		電話	
ご住所	都・道 府・県		

職業：小学生／中学生／高校生／高専生／短大生／大学生／専門学校生／予備校生／アルバイト／公務員／会社員／その他（　　　）

本誌を 毎月・時々・初めて 読む　何人で読みますか（　）人

自分のホームページを　持っている・持つ予定・持っていない・予定なし

本誌のほかによく読む雑誌は？［　　　　］

ベーマガお楽しみプレゼント⑥応募欄（134ページ掲載）	希望番号	品名
	番	


VM JAPAN
VANTAGE MASTER - MYSTIC FAR EAST
Falcom®
NIHON FALCOM CORPORATION
〒190-0012 東京都立川市曙町1-14-13
☎042(527)6501 http://www.falcom.com/
「VM Japan」の最新情報はファルコムホームページをご覧ください。
http://www.falcom.co.jp/
■社員募集(新卒・既卒・中途)！詳しくはホームページ「会社案内・社員募集」をご覧ください。
2002年6月発売予定！



本能はワイルドに、感情はディープに。AIBO ERS-220起動。

商品およびご購入に関するお問い合わせ
□AIBOインフォデスク 0570-008844［ナビダイヤル:全国どこからでも市内通話料でご利用いただけます］□携帯電話・PHSから 03-5470-0660［受付時間:午前10時～午後6時(年末年始を除く)］
□インターネット http://www.aibo.com/［ネットワークやサーバーなどのメンテナンス期間を除き24時間受付］
エンターテインメントロボット"AIBO"ERS-220 希望小売価格 180,000円(税別)
□AIBO本体のみでは基本的な動作しかできません。より個性的な動きをAIBOに組み込むためには別売ソフトが必要です。□商品の色は実際とは若干異なる場合があります。□"AIBO"および"AIBOロゴ"は、ソニー(株)の商標または登録商標です。※背景とAIBOは合成によるものです。実際の商品の大きさとは異なりますのでご注意ください。また、屋外では使用しないで下さい。製品に傷がついたり不具合の原因となることがあります。

第二十一巻第六号通巻二四〇号 昭和五十七年十二月六日第三種郵便物認可 二〇〇二年五月二十五日印刷 二〇〇二年六月一日発行
電波新聞社 〒141-8715 東京都品川区東五反田一―十一―十五 電話〇三(三四四五)六一一一

定価550円 本体524円 雑誌18361-06 T1118361060553

2002

わかる！動かせる！
プログラムが組める雑誌

マイコンBASIC Magazine

6

特集
携帯電話で動くプログラムに挑戦！

君にも作れるiアプリ

http://www.basicmagazine.net/

電波新聞社